EVALUATION OF DETERMINISTIC MODELS FOR NEAR SURFACE SOIL MOISTURE PREDICTION

Final Technical Report

by

J.E.Cochrane and M.G.Anderson

May 1988

European Research Office
United States Army
London England

CONTRACT NUMBER DAJA45-85-C-0007

Dr. M.G. Anderson

Approved for Public Release; Distribution Unlimited

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

A196 2

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188
Exp Date Jun 30, 1986

1a REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4 PERFORMING ORGANIZATION REPORT NUMBER(S)

5 MONITORING ORGANIZATION REPORT NUMBER(S)
R&D 4216-EN-01

6a. NAME OF PERFORMING ORGANIZATION
University of Bristol

6b. OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION
USARDSG (UK)

6c. ADDRESS (City, State, and ZIP Code)
University Road
Bristol BS8 1SS, UK

7b. ADDRESS (City, State, and ZIP Code)
Box 65
FPO NY 09510-1500

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
USAE Waterways Experiment Station

8b. OFFICE SYMBOL (If applicable)
WES-CE

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
DAJA45-85-C-0007

8c. ADDRESS (City, State, and ZIP Code)
PO Box 631
Vicksburg, MS 39180-0631

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO
61102A	1L161102BH57	01	

11. TITLE (Include Security Classification)
(U) Evaluation of Deterministic Models for Near Surface Soil Moisture Prediction

12. PERSONAL AUTHOR(S)
J. E. Cochrane and M. G. Anderson

13a. TYPE OF REPORT
Final

13b TIME COVERED
FROM Feb 85 TO Apr 88

14 DATE OF REPORT (Year, Month, Day)
March 1988

15. PAGE COUNT
316

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
08	13	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The objective of the Bristol Off-Road Trafficability Scheme (BORTS) is to develop an operational system to predict off-road trafficability on a km^2 scale for application to route management problems. The system design only allows the use of readily available data and highlights user friendliness as an important facet of the software technology. This new predictive technology is based on well established physical hydrological and evaporational processes along with a newly developed soil strength - soil moisture physical relationship. The Variable Source Simulator 2 (VSAS2) is available for use within the scheme. A route management scheme demonstrates the utility of the scheme and allows initial comparisons between the BORTS and the established U.S. Corps of Engineers system for predicting soil moisture and soil strength (SMSP model).

The system development, use of the system, limitations, initial comparisons with the SMSP model and future development are described in this report. The computer program has been written in FORTRAN 77 and is listed in the appendices along with a code glossary.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☒ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Jerry C. Comati

22b TELEPHONE (Include Area Code)
+441-402-7331

22c OFFICE SYMBOL
AMXSN-RE

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

88 6 27 001

ACKNOWLEDGEMENTS

The authors acknowledge the support and discussions with John Collins of the Waterways Experiment Station (WES) throughout the duration of the research. SMSP runs were undertaken by the Mobility Group at WES. In addition, Dr. George Baladi of WES provided assistance with certain of the initial ideas concerning soil strength/CI relations.

Requests for the disk in this report may be
referred to: U. S. Army Engineer Waterways
Experiment Station, ATTN: WESEN-C, P. O. Box
631, Vicksburg, MS 39180-0631
Per Mr. John Collins, USAEWES/WESEN-C

3 SOIL MOISTURE MODELLING

4 IMPROVED BSSS MODEL COMPONENTS

5 ROUTE SELECTION PROCEDURES

6 PROGRAM DESIGN

10 DISCUSSION

APPENDICES

ACRONYMS USED IN THIS DOCUMENT

BNGS	-	Bristol New Grid Scheme
BORS	-	Bristol Optimum Route Scheme
BORTS	-	Bristol Off-Road Trafficability Scheme
BSSS	-	Bristol Soil Strength Scheme
EEGS	-	Edit Existing Grid Scheme
SMSP	-	Soil Moisture Strength Prediction model
VSAS2	-	Variable Source Area Simulator 2

1. INTRODUCTION

1.1 Background

The demand for predicting field work conditions or off-road trafficability comes from agricultural, civil and military sectors. The spatial and temporal resolution of predictions is dependent upon application. For instance, the prediction of off-road conditions across an area of 200 km^2 in 20 days time for the passage of a large number of vehicles will be very different from predicting how soon those same vehicles can cross an area of 10 km^2 in the next 36 hours without being bogged down.

At the present time there are three main types of models used for predicting off-road conditions for the specific application to vehicle behaviour:-

1.1.1 Empirical Agricultural Models

Within agriculture there is a need to consider the degree of compaction a given type of vehicle may cause on a particular soil. Empirical relationships have been developed (Raghavan and McKyes, 1978) which for four soil types predict soil density (considered to be the dependent variable):

= f(moisture content, tyre slip, contact pressure, number of passes, depth and distance)

Further studies (Rhagavan et al., 1979; Huck et al., 1975; Taylor, 1971; Campbell, 1982) show that it is relatively easy, if demanding of data, to extend the analysis into an examination of traffic-soil-plant relationships. This will be of great importance in predicting plant root damage caused by the operation of agricultural vehicles.

Studies by Dyer (1980) based on the empirical Versatile Soil Moisture Budget - Version Three, linked with historic meteorological data, attempts to predict workday information across Canada. This scheme is reviewed in more detail in **section 2.3**.

1.1.2 Empirical Military Models

A review of soil trafficability prediction undertaken by the U.S. Corps of Engineers (1967) highlighted soil moisture - soil strength relationships (e.g. Collins, 1967; Moltham, 1967). The estimation of these two environmental variables and their inter-relationship is central to these tyᵥe of trafficability models. These models based the trafficability predictions through the calculation of the rating cone index (RCI) relationship developed by Collins (1971) and is defined by:

$$\ln RCI = 4.605 + \frac{2.123 + 0.008(C) - 0.693 \ln M}{0.149 + 0.002(C)} \qquad \textbf{eq. 1.1}$$

where M = moisture content (% dry weight)

C = percentage clay

RCI values were established, below which specified vehicles could not complete more than 50 passes. These RCI values were known as the vehicle cone index (VCI_{50}) for the vehicle and therefore predictions of RCI could be related to vehicle mobility.

To date the U.S. Army Corps of Engineers has utilized the soil moisture strength prediction model (SMSP) developed by Smith and Meyer (1973) to predict RCI. The SMSP model predicts soil moisture on a daily basis from empirical bookkeeping relationships, with seasonal constraints which then link with empirical relationships in the form of **equation 1.1**.

There is also available models predicting the effect of terrain on vehicle mobility and part of their input requirements include RCI. These models include the Army Mobility Model (AAM); the DMA/ETL Cross-Country Movement (CCM); and further improvements include a computerized condensed AMM system (CAMMS) by Turnage and Smith, 1983. The SMSP model and its linking with mobility models is discussed in more detail in **section 2.2**.

1.1.3 Deterministic Models

The use of the mechanical properties of the soil, i.e. soil density, cohesion and angle of internal friction, in predicting cone index has been developed by Rohani and Baladi, 1981. The inclusion of moisture content could lead to a physically based alternative to **equation 1.1**. A deterministic off-road trafficability prediction model does not at this stage exist though a physically based infiltration scheme combined with the empirical relationship give by **equation 1.1** has been developed by Anderson, 1983, and is described in **sections 3.2, 3.4** and **4.1**. A physically based infiltration scheme developed by Clapp, 1982, is reviewed in **section 2.5** and assessed for its potential to solve the requirements set out in **section 1.2.**

The desire for a new system is either because there is no existing system or there is a major problem or restriction with the old system. The three types of trafficability models discussed above, and the schemes which exist indicate that application of off-road trafficability studies have always been on a long time base, i.e. output from the empirical models is always on a daily basis. The scenario set at the beginning of this section where information on how soon vehicles can cross a given area in the next 36 hours is not catered for. The only attempt at a variable temporal resolution model is that of Andersons' (1983) and it is this model that is further developed in this project.

It is evident that a new physically based system is necessary which has variable temporal and spatial resolution, firstly because there is no satisfactory existing system, and secondly, because there are major restrictions with the old empirical schemes. The first step in the development of a project is to set out what the project aims to do (**section 1.2**) and the strategy adopted (**section 1.4**) to fulfil those aims.

1.2 Project Objectives

The requirements of any project may be defined through analyzing the problem to determine its nature. A requirement is a feature of the system, or a description of a facility the system is capable of doing in order to fulfil the system's purpose. The main purpose of this project is 'to develop an operational system to predict off-road trafficability on a km^2 scale for application to route management problems'.

The following general requirements are the first stage in the overall strategy for the project's development. The requirements are also shown in **figure 1.1** under various categories of requirement type.

1.2.1 General Requirements

1.2.1a To develop an operational system to predict off-road trafficability on a km^2 scale.

1.2.1b To develop a deterministic off-road trafficability model, and hence one requiring no calibration.

1.2.1c To construct a predictive scheme with variable temporal resolution.

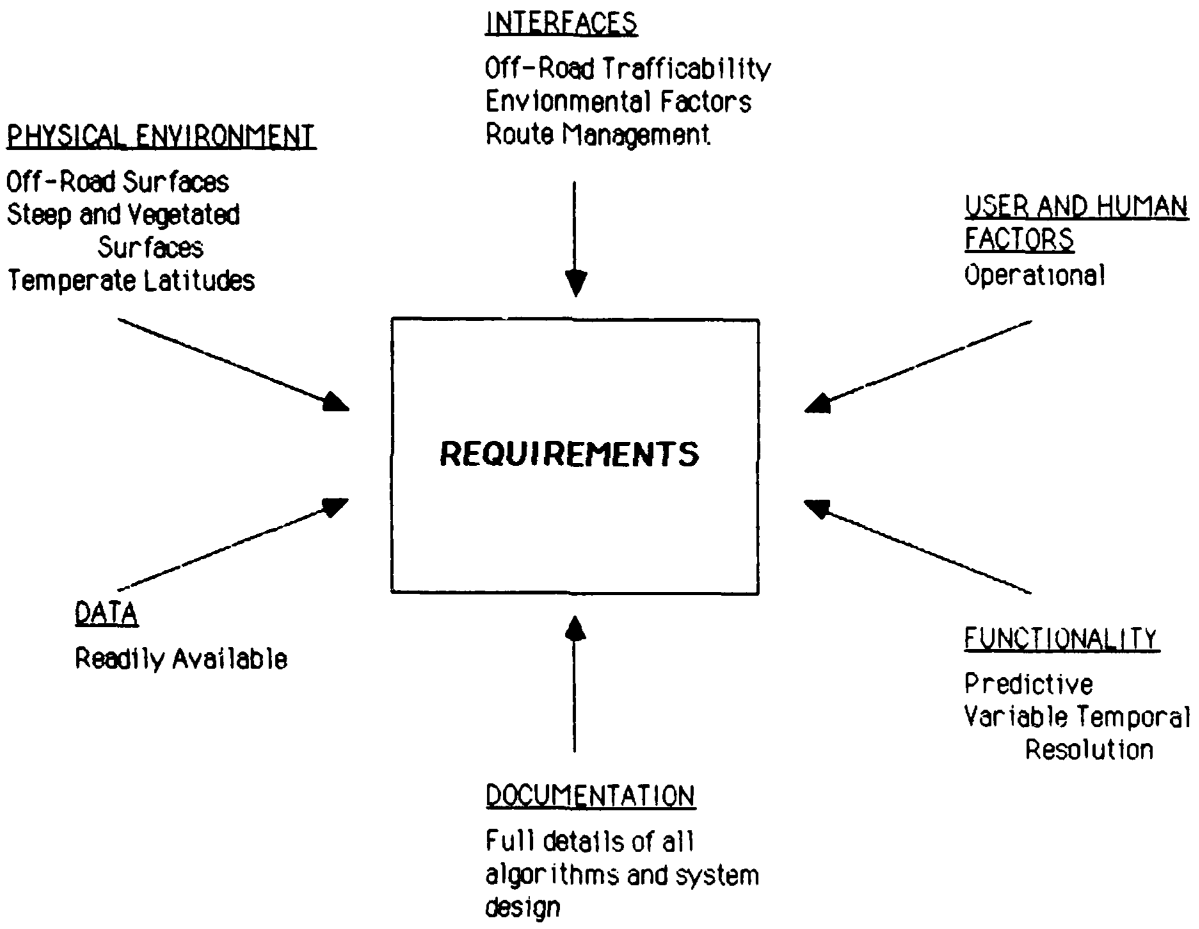
1.2.1d To develop a physically based system for predicting various environmental factors; e.g. soil moisture and soil strength.

1.2.1e To develop a facility capable of handling a steep area that may be highly vegetated.

1.2.1f To develop a system in-which the predicted off-road trafficability can be applied to route management problems.

1.2.1g To develop a system requiring only readily available input data.

FIGURE 1.1 : REQUIREMENTS ANALYSIS



The general requirements listed above will be further refined in **sections 3,4**,and **5**. The concept of requirement definition linked with a top-down approach is discussed in **section 1.4.1**.

The design of the entire BORTS system (**figure 1.3**) has been based on the definition of the general requirements and the subsequent refinement of these requirements in **sections 3,4** and **5**, plus the development strategy outlined in **section 1.4**. The large range of the requirements have led to two models for predicting off-road trafficability. These are the Bristol Soil Strength Scheme (BSSS) which solves the specific requirements **3.2.1, 3.4.1, 4.1.1, 4.2.1** and **4.3.1**, and an adapted scheme - the Variable Source Area Simulator (VSAS2) solving the specific requirement **3.3.1**. The components of these two models are described in **sections 3** and **4**. **Section 5** describes the application of off-road trafficability to route management.

This section is represented in **figure 1.2** as stage 1 - a problem was posed and the needs determined.

FIGURE 1.2 :STAGES OF SYSTEM DEVELOPMENT

STAGE	QUESTION	ACTION
1 Requirements definition	What is the problem?	Determine needs.
2 System design	What is the solution?	Design a solution to solve a perceived problem.
3 Program design	What are the methodologies that best implement the solution?	Specify algorithms and data structures to implement the solution.
4 Program implementation	How is the solution constructed?	Write code that implements the design and integrates code modules into a working system.
5 Testing	Is the problem solved?	Test code and system.
6 Delivery	Can the solution be used?	Write documents to describe the system. Train users and operators.
7 Maintenance -	Are enhancements needed?	Modify and enhance the system as necessary or desirable.

1.3 Originality

As stated in **section 1.1** the desire for a new system is either because there is no existing system or there is a major problem with the old system. In this project the U.S. Corps of Engineers SMSP model is considered as the 'old' system and is discussed in **section 2.2**. It is considered unsatisfactory for several reasons, and these may be summarized as follows:-

1.3a the scheme is entirely empirical and therefore inflexible with respect to temporal resolution,

1.3b the scheme is inflexible with regards attempts to up-grade, and therefore cope with phenomena such as hourly variations of evaporation,

1.3c the data required to run the scheme includes empirical relationships that are difficult to evaluate especially for the non-expert, and

1.3d results from the SMSP indicate some peculiarities when predicting RCI values from initial moisture categories of very wet and very dry. Tracing these peculiarities in an empirical scheme will be a difficult job.

Having realized the restrictions of the 'old' scheme it should also be pointed out that strictly the SMSP scheme is very different from the system proposed in **section 1.2**. In fact, it could almost be said that the only scheme approaching the requirements set out in **section 1.2** is that of Anderson's (1983). It is therefore proposed to develop a new system based on Anderson's model but with the following new aspects:-

1.3e new deterministic, physically based soil moisture - soil strength routine,

1.3f corrected and improved evaporation routine with respect to inclusion of albedo,

1.3g improved data requirements - data bank development, and

1.3h user friendliness operation of system.

The project requirements have also led to the new development of the following:-

1.3i the adaptation of an existing scheme to deal with steep slopes, convergence of flow and highly vegetated areas,

1.3j the development of a route management scheme,

1.3k the application of the scheme to existing mobility models, e.g. the Ministry of Defence's DRIVEB model,

1.3l the development of a system that is menu driven, system prompted and with basic system checks,

1.3m the development of documentation and code that will allow easy maintenance and alterations by new developers, and

1.3n the formulation of guidelines on which scheme to predict RCI should be used under which environmental conditions.

This is a first attempt to develop such a system, and as such will provide an insight into the problem area. The development and initial results have produced a well defined directive for the project after this development period (**section 10**).

1.4 Development Strategy

The development of any computer system requires the adoption of a logical, clearly defined strategy which incorporates a progressive building block technique. This is represented in **figure 1.2** as a series of seven work stages. As shown in **figure 1.2**, each stage is described by a titl-, a question and the action necessary at that specific stage to solve the problems. Once each stage has been completed, the developer can start the next stage - though often it is necessary to return to a previous stage for re-assessment and adjustment of the solution. All adjustments and their interrelationships with other elements of the solution must be traced through all subsequent stages. The following section describes the stages shown in **figure 1.2** in more detail.

1.4.1 Stage One: Requirements Definition

Prior to the design of any system the developer must understand the system's purpose. This starts with an examination of requirements (**section 1.2**). The requirements of any project may be defined through analyzing the problem to determine its nature. A requirement is a feature of the system, or a description of a facility the system is capable of doing in order to fulfil the system's purpose.

Throughout this project a 'top-down' approach has been adopted. With respect to the requirements of the project this means that initially the requirements or objectives have been expressed at the very highest level in general terms (**figure 1.1**); then at subsequent levels the requirements are made more specific (**sections 3,4 and 5**). For example, high level requirement **1.2a** states that the system should be 'operational', lower levels of definition restate the requirement in specific terms of parsimonious data requirements, readily available data, and menu driven system operation.

In this requirements analysis, we have determined exactly what problems need a solution. The requirements have been satisfactorily specified and understood by the developer, now the next stage of system design can begin.

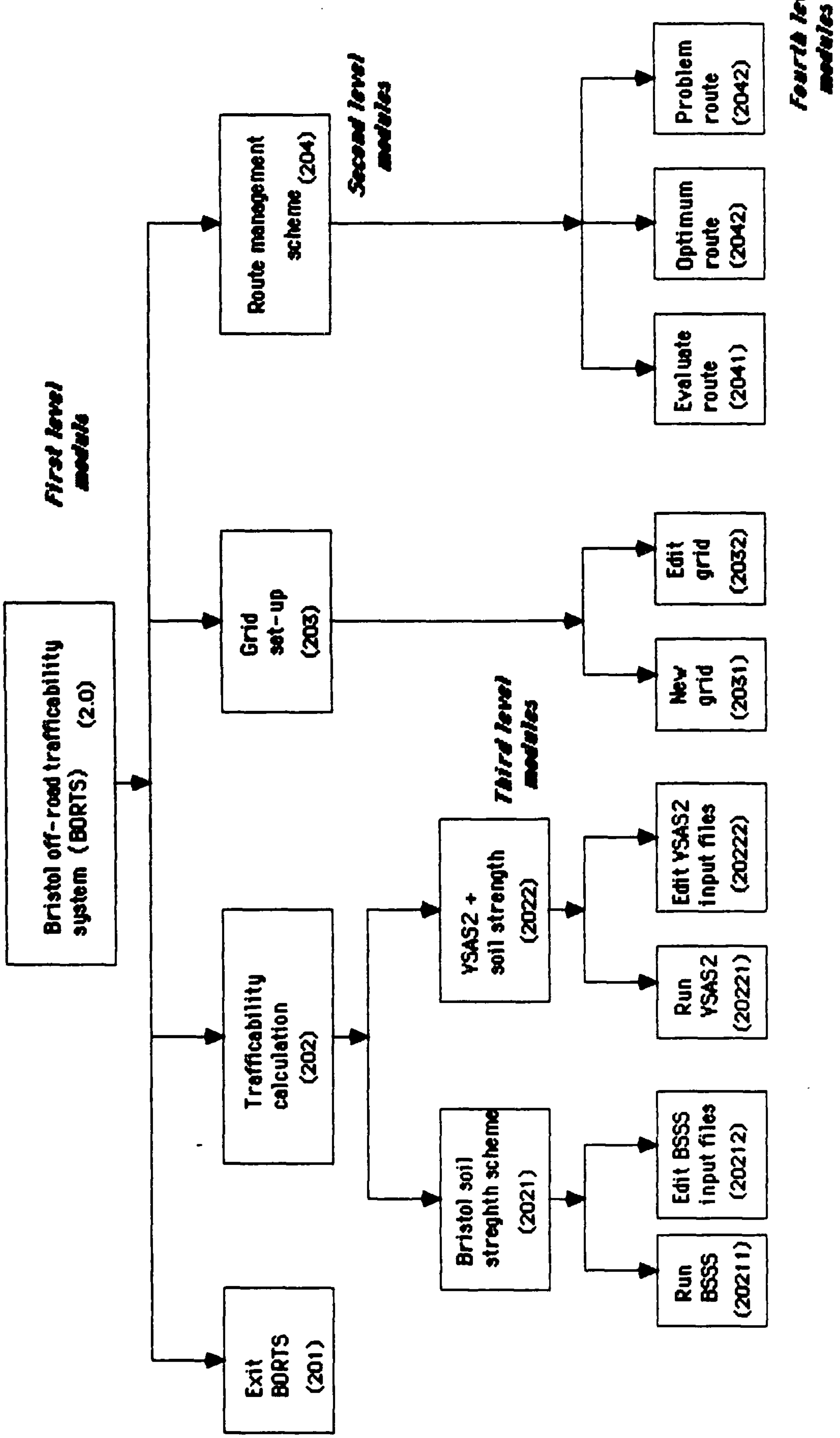
1.4.2 Stage Two: System Design

The specification of the system requirements allows the developer to translate those requirements into a system that will satisfy the project's needs. It is this translation of the problem into a description of the solution that is known as system design.

For each component of the system there may be more than one solution. An assessment of existing and specially devised solutions is a necessary sub-stage of the system design. The development of the system design in **section 6** follows on logically from the technical descriptions of the major system components in **sections 3,4** and **5**.

A top-down approach has also been adopted to aid the process of system design. As shown in **figure 1.3** the design has been separated into composite parts known as design modules. Each module is a working entity having specified sets of input and output. Working in a top-down manner the design should ensure that all information entering the system is completely defined; all functions are specified; and that output from the system is that required by either the user or other systems. The fourth

FIGURE 1.3 DESIGN CONCEPT



level, or bottom level, modules shown in **figure 1.3** are where the actual functions are carried out and these are detailed in **section 6** along with information on how the components link together in each module and how the modules interrelate.

The completion of the system design, that is, an explanation of exactly how the system will work in terms of data flow and transformation, allows the developer to translate it into a program design.

1.4.3 Stage Three: Program Design

The system design does not allow the developer to write program code directly from its description. The design modules and their inter-relationships now have to be further specified in terms that are instructions for the programmer. The program specifications have been written in pseudo code' as an intermediate stage between the English system design description and the program code. **Section 6** details the program design and tabulates the pseudo code for each algorithm to be included in each design module.

1.4.4 Stage Five: Program Implementation

The design is now ready to be translated into program code which functions correctly and preserves the desirable design characteristics. The characteristics included in the system design and refined in the program design (such as distinct modularity, well defined interrelationships etc.) should be inherent in the program or programs written, so that all algorithms, subroutines, functions and data structures can be easily traced from the system design to program implementation and vice versa.

The code has been written (**Appendix B**) in small separate subroutines all of which are cross-referenced with the program design in **section 6**. A system of naming each subroutine according to the design module it is a component of has been adopted to make the task of future development a much easier task. It should be remembered that the *general purpose* of the system being developed in this project is unlikely to change throughout the software development period and for sometime afterwards. However, modifications and enhancements in the *nature* of the system might be necessary. This programming style or manner easily allows the addition, replacement or up-grading of subroutines with ease and suggestions for improvement are discussed in **section 10**.

The programs now coded in FORTRAN (**Appendix B**) require to be tested to see whether they work as intended.

1.4.5 Stage Five: Testing

The testing of the programs is split up into several sections and the methodologies of testing used in this project have been:-

1.4.5.1 *Defective software*, where there is a software error. This usually means that the software does not do what the requirements specified. These errors must be traced, and are normally of the following types:

(a) the requirements specification may be wrong, i.e it is not really what is wanted,

(b) the requirements specification may be computationally or physically impossible,

(c) the system design may be faulty,

(d) the program design may be faulty, and

(e) the code itself may be wrong.

1.4.5.2 *Unit testing*, where after the code has been examined for errors, each subroutine and then module is compiled and run with test data to search for other errors. Finally, the entire system is compiled and run with test data.

1.4.5.3 *Program review*, which explains in words, diagrams and technical terms what each program is supposed to do in code.

1.4.5.4 *Proving programs correct*, where through unit testing, a program is correct if it implements the specifications of the program design and interfaces properly with the other modules.

It is important to realize that no system is ever 'error free' and International Systems , Inc., Pennsylvania state in their "Laws of Project Management" that 'No system is ever completely debugged. Attempts to debug a system inevitably introduce new bugs that are even harder to find'

Test input data has been designed so that the output demonstrates something about the behaviour of the program (**sections 9** and **10**). To test a module the input data and conditions must allow the program to manipulate that data, and observe the output from the program. During initial development the system is often set-up to provide an abundance of extra intermediate output information to aid in locating software errors.

Comparisons have been made with an existing scheme as an initial methodology of model validation and are discussed in **section 10**. The entire system is now ready for stage six - that of delivery.

1.4.6 Stage Six: Delivery

The main component of the delivery of the system developed in this project is that of writing an effective documentation. This is aimed at two levels , firstly at the user, and secondly, at the developer or operator. This report covers both these aims by following the development strategy discussed in this section and outlined in **figure 1.2**.

As shown in **figure 1.2** the delivery also includes training users, therefore a 'hands on user guide' has been developed along with test data. This is detailed in **section 9**. Instructions for the operator are given in **Appendix C.**

1.4.7 Stage Seven: Maintenance

Section 10 discusses some initial results which in turn permits a prescription for further enhancements to the system to be written. The code description, program design and system design detailed in this report should help in the correction of errors, inclusion of new components, perfection of existing components and of course prevention of future failures.

1.4.8 Software and System Quality

The quality of software and systems is reflected in their characteristics. The characteristics required for high quality depend on who is assessing the system. A user will assess software to be of high quality if it does what they want it to do and is easy to learn and use. A designer and maintainer will look for a design and code that is easy to test and maintain. The characteristics considered to be important in the quality design of this project are as follows:

1.4.8a *Usefulness*: The software system must be useful. Firstly, it must do what is required of it. Secondly, it must be easy for the maintainer to locate the source of an error, find the modules where a particular function is coded, trace the code and modify and add other subroutines.

1.4.8b *Reliability*: The system must produce the correct result to the correct degree of accuracy.

1.4.8c *Accessibility*: The system must perform its functions in a timely manner and therefore data should be available when needed and the system should respond to the user in a reasonable amount of time.

1.4.8d *Human compatibility*: The users and developers must find the system easy to learn and use. This might be the most important characteristic of the system. A system that while performing its functions perfectly is impossible for the users to understand, can only be considered as a failure.

The development of the software system has attempted to follow the strategy outlined in this section and to make these characteristics of high quality software inherent within the structure.

1.5 Scope of this Report

The scope of this report is to document all the stages shown in **figure 1.2** and discussed in **section 1.4**. The background, general requirements, the developmental strategy adopted and originality of the project are all discussed in **section 1**. A review of existing schemes is addressed in **section 2**. The development of the new system is detailed in **sections 3, 4,** and **5**. A system description listing the system entities, attributes and their relationships with each other is given in **section 6**. Guidelines for the user are given in **section 7** and a 'hands on ' user guide is provided in **section 9**.

The restrictions and assumptions of the main components of the system are set out in **section 8**. These restrictions along with some initial results and a discussion of the development, leads to a written prescription for future development of the system in **section 10**.

The program code and test data are given in **Appendix B** along with compilation instructions for the operator in **Appendix C**. The definition of some of the terms used in this report are given in **Appendix A**.

1.6 Conclusions

Before defining the proposals for a new system it is vital to examine how successful existing systems are. Are users happy with their existing software and systems? Can the systems be extended to be used for slightly different applications? The answer is yes and no.

Often systems work, but not exactly as expected. For instance, the peculiarities encountered when the SMSP model has initial moisture categories of very wet and very dry but appears to work well for the other initial moisture contents (results from SMSP model, 1987, **section 10.3.1**).

The system may work for the first scenario given in **section 1.1**, where information regarding trafficability 20 days ahead is required. But, the second scenario, where information is required over the next 36 hours, at say hourly intervals, is impossible for the system to cope with.

Having decided that there is a gap in the capabilities of existing systems, a well defined set of requirements is drawn up. In the desire to produce a system of acceptable quality and utility an approach strategy has to be adopted. This has been described in **section 1.4**, though it is important to understand the difference between 'good' and 'bad' software.

The characteristics of software and system quality depends on who is analyzing it. A developer will assess in terms of the following:

1.6a How easy is it to design, code, test and maintain?

1.6b Is it efficient in terms of computer usage?

A user will assess in terms of the following:

1.6c Will it do what I want it to do?

1.6d Is it easy to learn and use?

Hence, it can be seen that high quality software and systems have characteristics that are addressed to the requirements of the users, developers and maintainers. The following report documents the development of the proposed system (**section 1.2**), and through the adoption of a logical clearly defined development strategy incorporate high quality for the users, developers and maintainers.

2. REVIEW AND ASSESSMENT OF EXISTING SCHEMES

2.1 Introduction

The prediction of soil moisture and soil strength is not a new concept. There are many models which attempt to predict soil moisture at prescribed temporal and spatial resolutions. There are also several models which link soil moisture and soil strength characteristics to predict soil strength. This section reviews four models, two of which were designed specifically for off-road trafficability prediction, while the other two, although designed for different purposes, could be linked with appropriate submodels to predict off-road trafficability.

The first model discussed is the 'Automatic Model for Predicting Soil Moisture and Soil Strength' (SMSP model) developed by Smith and Meyer (1973) for the U.S. Corps of Engineers (**section 2.2**). The SMSP model predicts daily soil moisture and strengths of soil layers 0-15cms and 15-30cms. It is an empirically developed model and is currently used by the U.S. Corps of Engineers for off-road trafficability prediction.

The second model discussed is the 'Versatile Soil Moisture Budget - Version III' (VB III) developed by Dyer and Mack (1984) for use by the Canadian Department of Agriculture (**section 2.3**). The VB III predicts daily soil moisture contents through the profile. By linking current status of the areas to be modelled with 'typical' or 'worst' scenario historic meteorological data, the model predicts the soil water status for large areas of agricultural land for up to three months ahead. Predictions are updated every week. It is an empirically developed model and is currently used by the Canadian Department of Agriculture for predicting drought and work day analysis used in their 'aid and advice' to farmers policy

The third model is the German terrain trafficability model developed by Hanl and Tries (1982) for evaluating trafficability of tanks with respect to weather influence (**section 2.4**). The Hanl and Tries model predicts daily soil moisture and categories of trafficability. It is an empirically developed model which through several important restrictions (discussed in **sections 2.4** and **2.6**) is simple to use.

The fourth model is the 'Wetting-Front Model' developed by Clapp (1982). The wetting-front model (**section 2.5**) was designed to predict the locations of wetting-fronts throughout a soil profile which might have layers with different soil characteristics. It includes a determination of evaporation from the soil surface which would be useful in trafficability applications. The model has been developed on deterministic relationships and the significance of this is further discussed in **section 2.6**.

The four models are outlined in terms of their design and input data requirements. They are evaluated (**section 2.6**) in terms of their ability to fulfil the objectives outlined in **section 1.2**. By reviewing these four models, it would be prudent to indicate that there are many other similar models and submodels of larger schemes which predict soil moisture, but it becomes apparent that to fulfil the general objectives in **section 1.2**, and the specific objectives detailed in **sections 3.2.1, 3.3.1, 3.4.1, 4.1.1, 4.2.1, 4.3.1, 5.2.1, 5.3.1** and **5.4.1**, that these four models are the closest to the proposed scheme to be considered as viable alternatives. The SMSP model is also important in that it is the working off-road trafficability scheme of the U.S. Corps of Engineers and comparisons with it will be used as a preliminary form of verification for the newly developed scheme.

2.2 Automation of a Model for Predicting Soil Moisture and Soil Strength (SMSP Model) Smith and Meyer, 1973

The SMSP model (**figures 2.1**and **2.2**) determines soil moisture using a bookkeeping procedure working on a daily basis. It is then linked with certain soil properties to predict soil strength in terms of cone index (CI) and rating cone index (RCI). Smith and Meyer determined through empirical investigation that the important soil layers concerning off-road trafficability were the 0-15 cms and 15-30 cms layers. The SMSP is empirical in nature and subsequently some input requirements are not always easy to determine. The terrain input requirements for the SMSP are of three different categories:-

2.2.a Specific data - unique characteristics and relationships for each site to be considered.

2.2.b Estimated data - averaged or estimated data from similar sites, field measurements and relationships developed from soil property data abstracted from a large number of sites.

2.2c Surface composition group data - similar to **2.2b** except that relationships are developed for a number of surface composition groups which are similar to the soil classes of the Unified Soil Classification System modified to permit characterization of the entire area.

As **figure 2.1** shows, the other major input requirements are that of daily precipitation, wetness index and the dates of the beginning of the different seasons (when rates of depletion change). The input data requirements are further detailed in **table 2.1** and used for the comparison and evaluation of the four models in **section 2.6**.

FIGURE 2.1 : GENERAL STRUCTURE OF THE SMSP MODEL

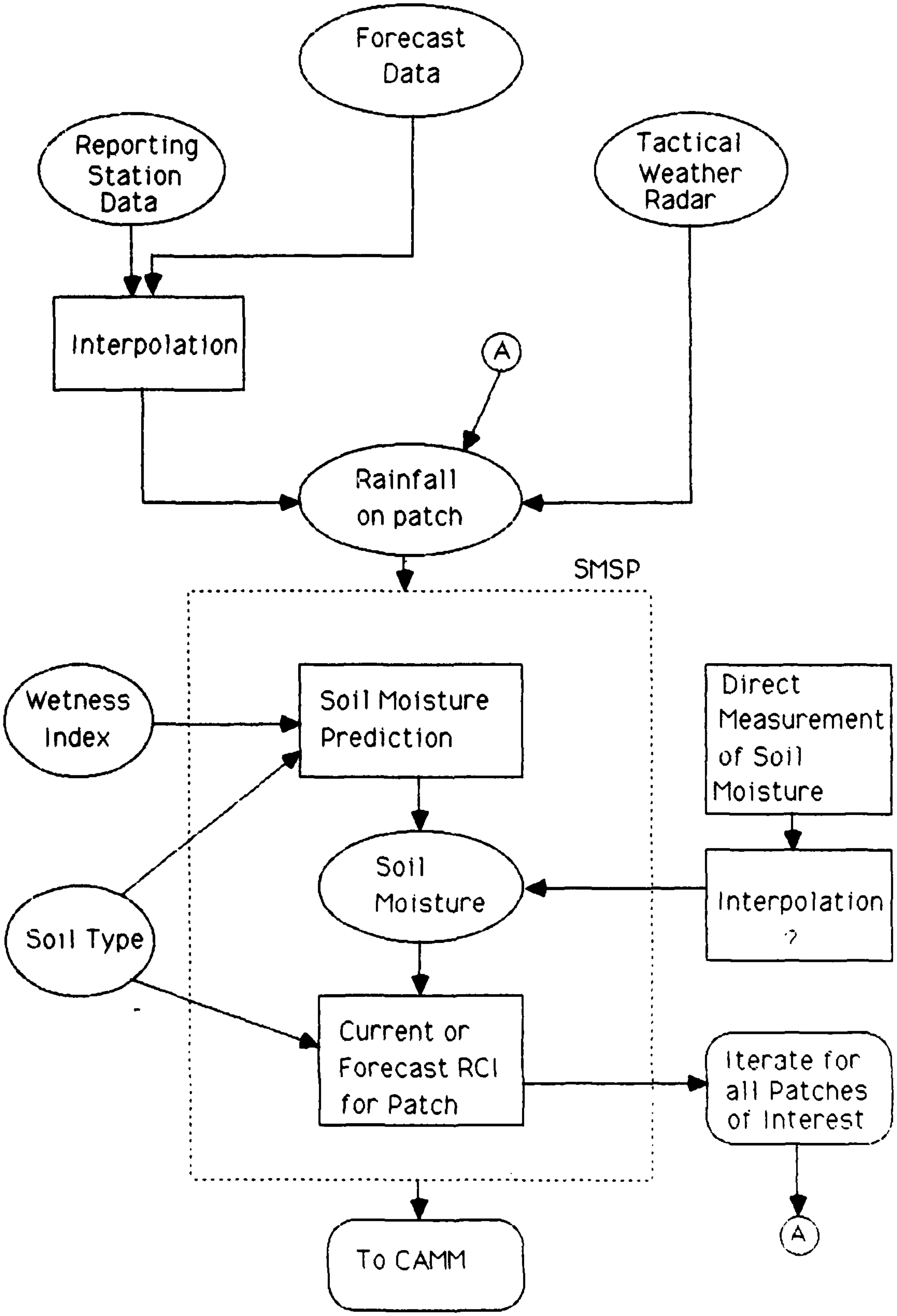


FIGURE 2.2 : SMSP MODEL

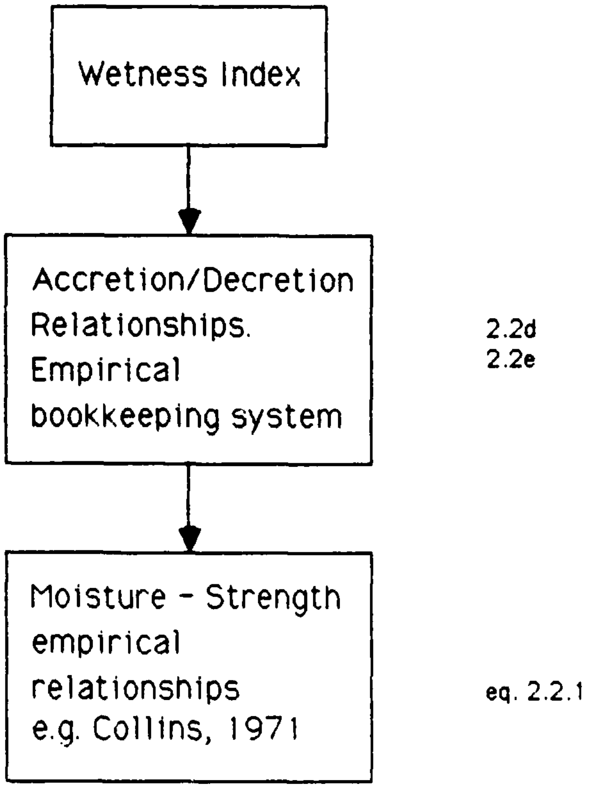

Wetness Index
Accretion/Decretion Relationships. Empirical bookkeeping system
2.2d
2.2e
Moisture - Strength empirical relationships e.g. Collins, 1971
eq. 2.2.1

TABLE 2.1 DATA REQUIREMENTS OF THE SMSP MODEL

(a) Site and Storm Parameters:

precipitaion (daily)
latitude
air temperature (daily)

(b) Soil Information:

soil texture
bulk density
initial soil moisture content
first layer soil moisture content storage for one year
second layer soil moisture content storage for one year
minimum precipitation causing accretion
maximum water content
minimum water content

(c) Coefficients:

for accretion equations
for tentative average depletion equations
for specific depletion relations
for moisture - strength relationship

(d) Constants and Factors:

for cone index relations
for maximum and minimum equations
for site depletion factor
for accretion equation

(e) Variables:

beginning day of data
beginning month of data
beginning year of data
number of days in each month
point at which accretion equation is modified
beginning day of new season
beginning month of new season
beginning year of new season

The governing relationships (**figure 2.2**) of the SMSP are simple, straight line and power functions derived empirically using regression techniques:-

2.2.d Accretion relations:

accretion = f (precipitation, amount of storage space available in the soil for absorbing water, soil layer in profile, season)

2.2e Depletion relations:

depletion = f (soil moisture content above a minimum moisture content, soil layer in profile, season)

The prediction of soil strength is based on the following relationship:

$$\ln RCI = a + b \ln(MC) \qquad \textbf{eq. 2.2.1}$$

where RCI = rating cone index

MC = soil moisture content (%)

a & b = user supplied constants from specific or estimated data

When considering surface composite groups the model assumes that a straight line relationship exists between ln RCI and ln MC. The equation for this straight line was established by forcing it through soil moisture contents at a RCI of 300 and CI of 200.

Output from the SMSP model is on a daily basis which may be over any part of the year (or several years) for one or more selected years of extreme, unique or typical rainfall distributions. There is no option for a shorter time interval between outputs, and any extension to a new site must include either specific or estimated relationships and input data. The application of the output information is in vehicle mobility models such as the Condensed army Mobility Model (CAMM) as shown in **figure 2.1**.

2.3 Versatile Soil Moisture Budget - Version Three (VB III) Dyer and Mack, 1984

The VB III model was originally developed from the simple water balance equations of Thornthwaite and Mather (1955) to the multi-layer budget of Holmes and Robertson (1959) to the versatile soil budget of Baier and Robertson (1966). The VB III model is the most recent improved version of the versatile budget and is directed at the agricultural sector. Estimates of daily soil water contents based on present and past weather events provide information to the Canadian farming community throughout the growing season and is the basis to field workday analyses (Dyer et al., 1978; Dyer, 1980)

The basic structure of the model is described by the flowchart in **figure 2.3** (from Baier et al., 1979). The soil moisture calculation within the cellular structure (**figure 2.4**) may be described by:

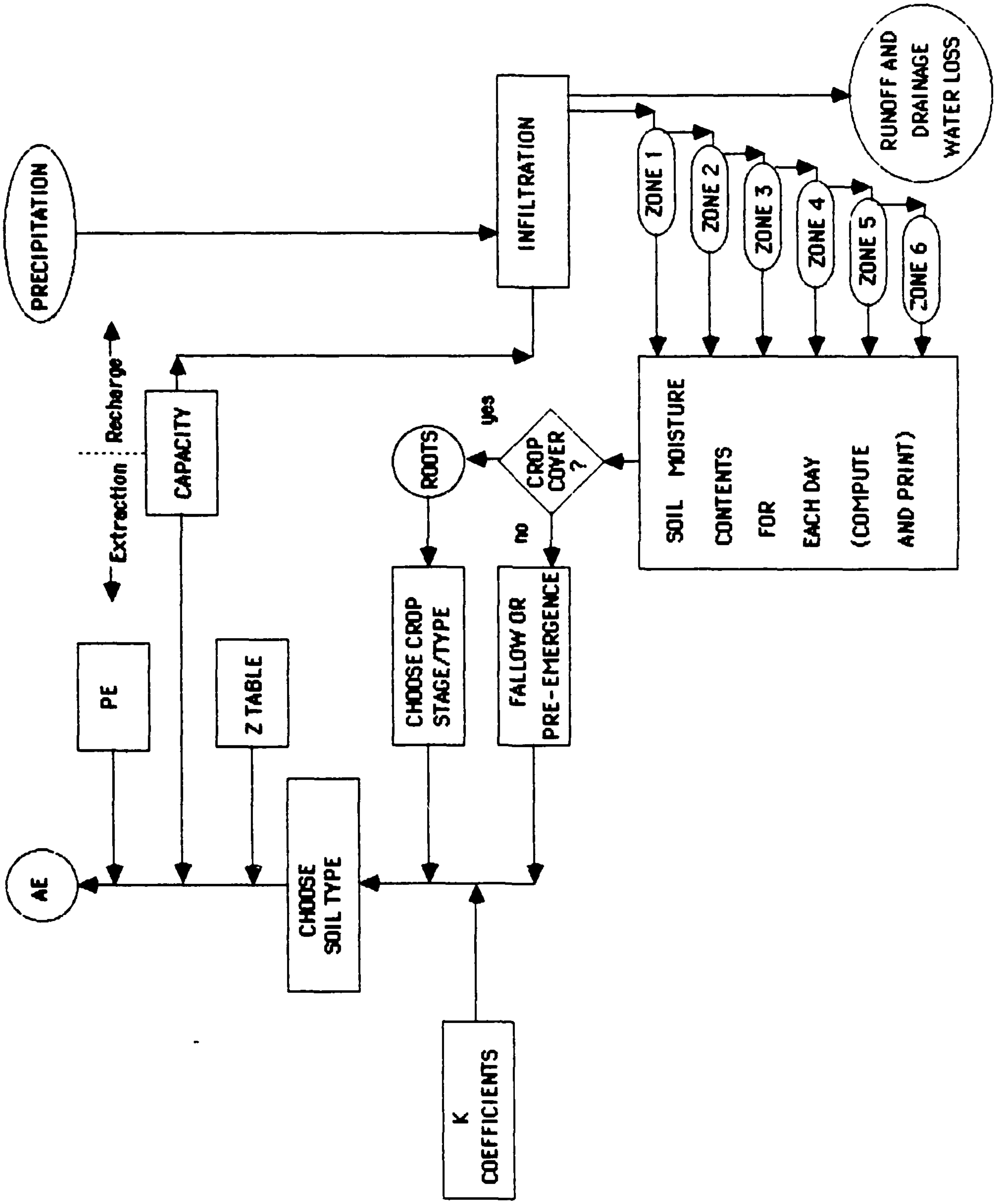
$$S1_i = S1_{i-1} - P_i - PDL_i - PDL_{i-1} - DF_{i-1} - DRN1_{i-1} - ASE_i \qquad \textbf{eq. 2.3.1}$$

$$S2_i = S2_{i-1} + DRN1_{i-1} - DRN2_{i-1} - AERT_i - DF_{i-1} \qquad \textbf{eq. 2.3.2}$$

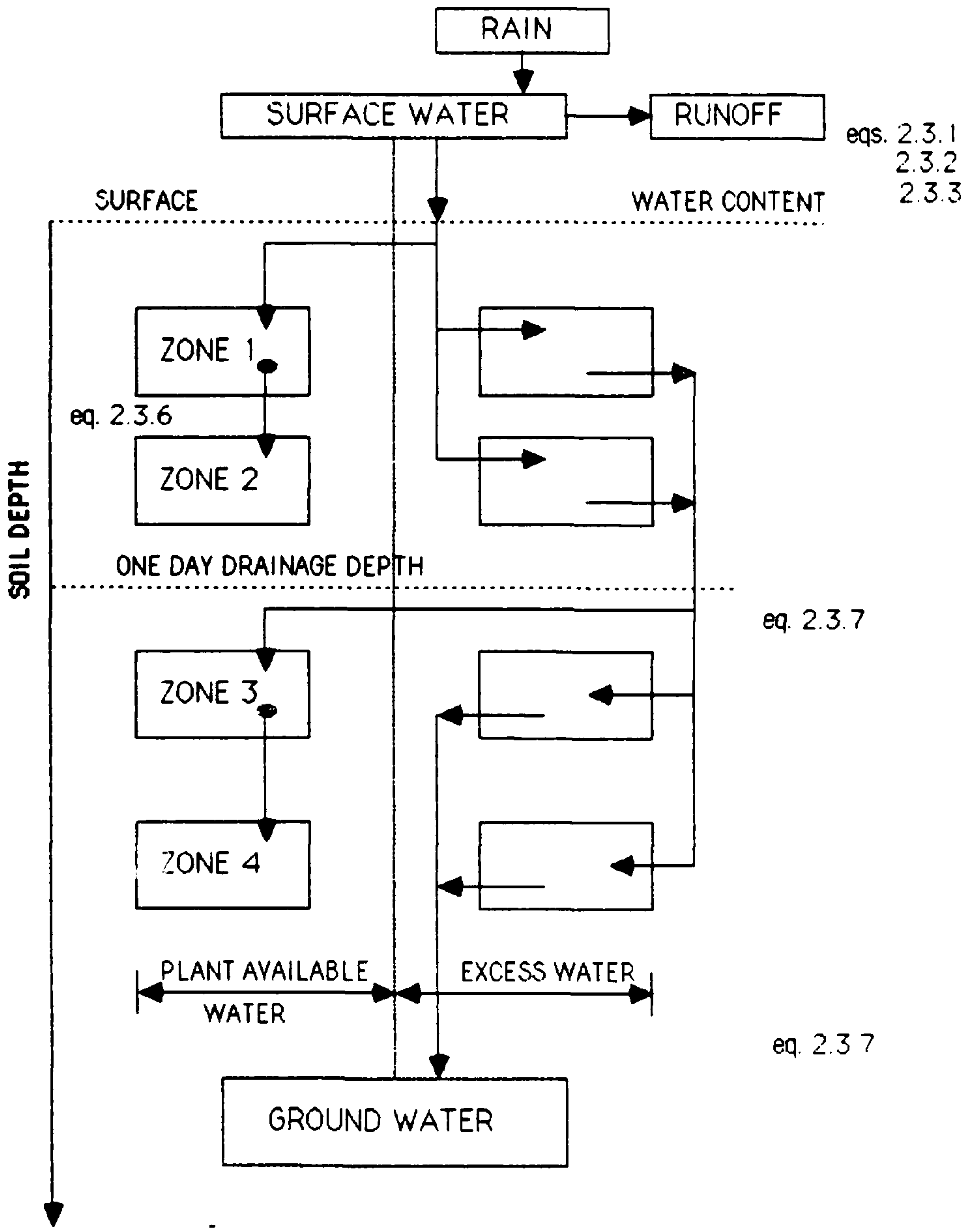
where S1 & S2 = plant available water in zones 1 and 2 respectively

P = daily rainfall

FIGURE 2.3 : FLOW CHART OF THE VB III MODEL



2.4 : CELLULAR STRUCTURE OF VB III



PDL = water left on surface

DF = water diffused between zones

DRN1 = downward drainage from zone 1

DRN2 = downward drainage from zone 2

ASE = actual surface evaporation

i = day number (julian)

Actual daily evapotranspiration (AE) may be described by:

$$AE_i = ASE_i + AERT_i \qquad \textbf{eq. 2.3.3}$$

Root extracted water (AERT) is given by:

$$AERT_i = RTX \cdot Z_2 \cdot PE_i \cdot (S2/C2) \qquad \textbf{eq. 2.3.4}$$

Actual surface evaporation (ASE) is given by:

$$ASE_i = (Z_i \cdot PE_i \cdot (S1/C1)) - AERT_i \qquad \textbf{eq. 2.3.5}$$

where C1 & C2 = available water capacities of zones 1 and 2 respectively

Z1 & Z2 = Z-table (drying curves) values to describe moisture retention for different soil types

RTX = root extraction coefficient $(1 > RTX > 0)$

The water diffused between zones is described by:

$$DF_i = \{(S2/C2) - (S1/C1)\} . RDC . C1 \qquad \textbf{eq. 2.3.6}$$

where RDC = redistribution coefficient $(1 > RDC > 0)$

Gravity water drainage out of zone 1 is given by:

$$DRN1 = (S1 - C1) . DRS \qquad \textbf{eq. 2.3.7}$$

where DRS = drainage coefficient $(1 > DRS > 0)$

N.B. $S1 > C1$

Drainage from zone 2 is assumed equal to the drainage into zone 2 from zone 1 on the previous day:

i.e. $DRN2_i = DRN1_{i-1}$

N.B. $DRN2 < (S2 - C2)$ and $S2 > C2$

From examination of these relationships it becomes apparent that there are many control coefficients required as input by the user (**table 2.2**). Although the VB III model has reduced the input requirements of the previous versatile budgets, its empirical and semi-empirical relationships demand much site specific information. The input detailed in **table 2.2** is used in the comparison and evaluation of the four models in **section 2.6**. There is no option for shorter time intervals between output.

TABLE 2.2 : DATA REQUIREMENTS OF THE VB III MODEL

(a) Site and Storm Parameters:

precipitation (daily)
potential evapotranspiration
crop stage dates (julian days)
julian calendar day

(b) Soil Information:

number of cells
cell thickness
cell field capacity
cell permanent wilting point
bottom cell depths
number of cells in layers 1 and 2
maximum drainage amounts (daily)

(c) Coefficients:

for soil water retention curves (Z - curves)
for each growth stage and cell
for fractional drainage out of layer 1

2.4 A Method for Evaluating terrain Trafficability of Tanks with respect to Weather Influences (Hanl and Tries, 1982)

The Hanl and Tries model attempts to use the established 'Terrain Trafficability Chart for Cross Country Movement' (CCM) of the Military Geographical Service in a more quantitative and rigorous manner. The model requires three types of input data:-

2.4a long term factors:- soil type, configuration of terrain,
2.4b medium term factors:- vegetation, seasonal cultivation of topsoil,
2.4c short term factors:- soil moisture content, precipitation.

Table 2.3 details the input requirements of the Hanl and Tries model for comparison and evaluation of the four models in **section 2.6.**

The model is based on a soil moisture balance account system (**figure 2.5**). A 'Soil Moisture Analysis Chart' is created through the continuous extrapolation and reckoning of soil moisture gain and loss. The particular form of the the budget run, i.e. gain of soil moisture content (**figure 2.6**) or loss of soil moisture content (**figure 2.7**), is dependent on whether rainfall exceeds the potential evaporation or not. The budget is run from 0600 hrs - 0600 hrs to output a daily trafficability map.

The empirical relationships given in **figures 2.6** and **2.7** utilise much site specific data (**table 2.3**) although the use of one standard reference soil ensures that this is an economic system to use. There is no option for shorter time intervals between output or to incorporate any rain between 0000 hrs and 0600 hrs in the trafficability prediction for the current day.

FIGURE 2.5 : FLOW CHART OF THE HANL AND TRIES MODEL

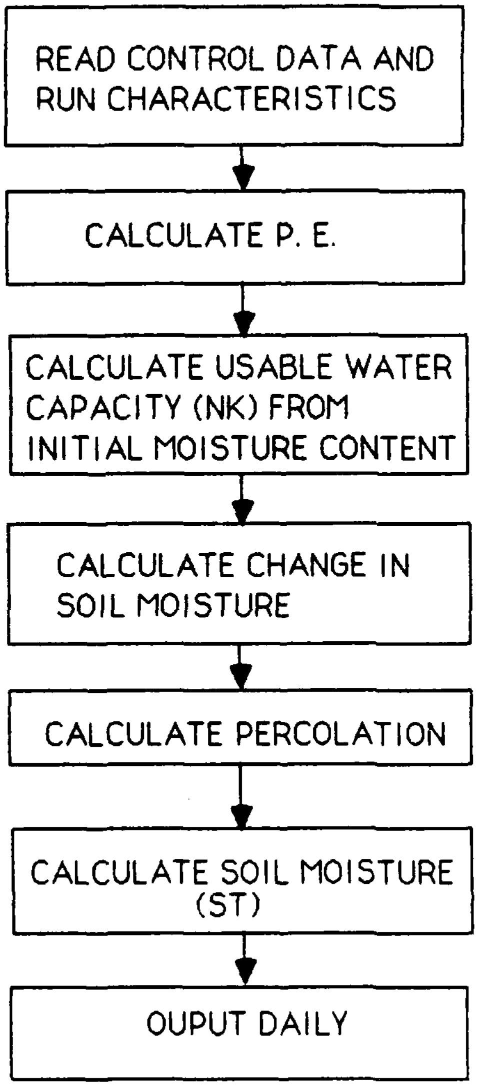

READ CONTROL DATA AND RUN CHARACTERISTICS
CALCULATE P. E.
CALCULATE USABLE WATER CAPACITY (NK) FROM INITIAL MOISTURE CONTENT
CALCULATE CHANGE IN SOIL MOISTURE
CALCULATE PERCOLATION
CALCULATE SOIL MOISTURE (ST)
OUPUT DAILY

FIGURE 2.6 : SOIL MOISTURE GAIN SITUATION OF THE HANL AND TRIES MODEL

Principle: Daily Balance Account of Loss and Gain of Soil Moisture Content

Basis: Climatic Balance of Water BIL = N - PE

Soil Moisture Content Gain N > PE

ΔSt = BIL = N - PE

1. BIL ≥ 0 (N ≥ PE, AE = PE)
when soil moisture content
ST ≥ 55% NK

2. BIL ≥ 0
when soil moisture content
ST < 55% NK

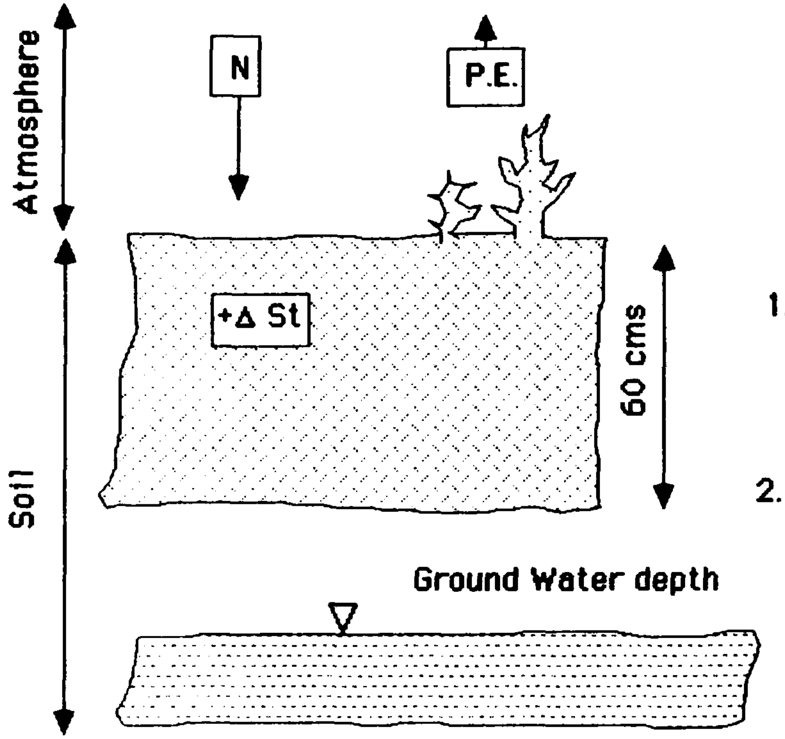
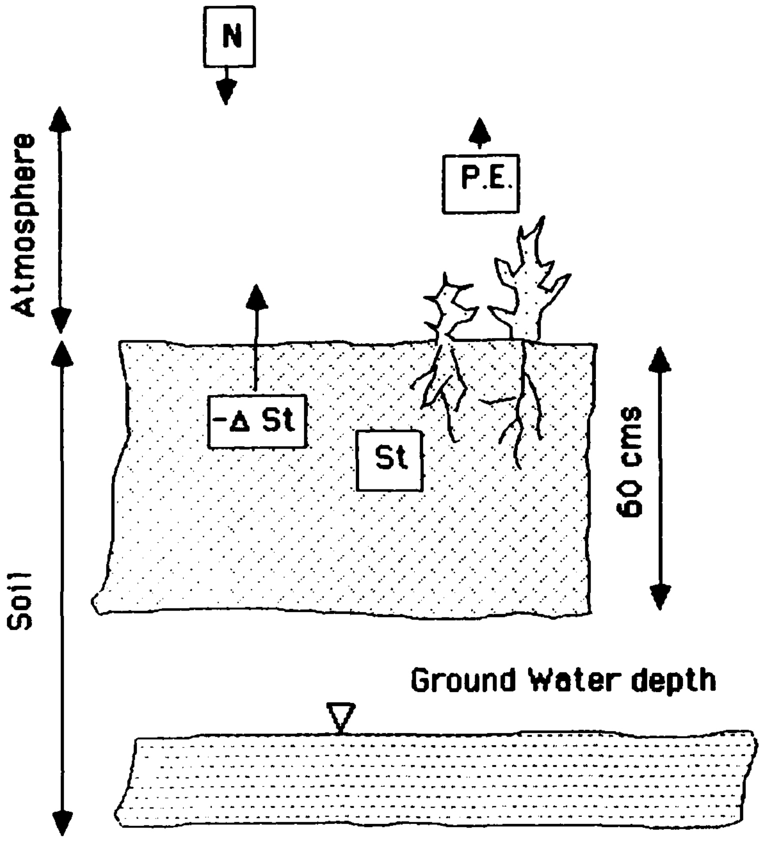


FIGURE 2.7 : SOIL MOISTURE LOSS SITUATION OF THE HANL AND TRIES MODEL

Soil Moisture Content Loss



N < PE

PFAU Formula:
$\Delta St = St_0 - St$

$St = St_0/(e**(A/NK))$

St = soil moisture content at the beginning of the day to be balanced

St = soil moisture content at the end of the day to be balanced

$A = |N - PE|$

NK = normal usable capacity

TABLE 2.3 : DATA REQUIREMENTS OF THE HANL AND TRIES MODEL

(a) Site and Storm Parameters:

precipitation (daily)
dry and wet bulb temperatures

(b) Soil Information:

initial soil moisture content
field capacity
wilting point
usable water capactity

(c) Coefficients:

Haude (ß) for calculation of P.E.

(d) Empirically derived look-up tables for:

relating soil types with standard reference soil terrain trafficability with respect to :-
(i) soil type and moisture state,
(ii) driving manoeuvres
(iii) terrain configuration and surface state

2.5 A Wetting-Front Model of Soil Water Dynamics (Clapp, 1982)

The wetting-front model aims to simulate the following processes in a vertical soil column:

2.5a one-dimensional infiltration described by the Mein-Larson / Green-Ampt relationships,

2.5b redistribution after infiltration,

2.5c evaporation according to an analytical diffusion equation using an explicit approximation to the mean weighted diffusivity (Clapp, 1982),

2.5d internal drainage.

The soil profile is represented as a series of uniform moisture blocks separated by wetting fronts (**figure 2.8**). The flowchart in **figure 2.9** shows that calculations of flow (Q_i), velocity of the boundary Z_i (Z'_i) and fluxes at the upper and lower boundaries of each block (Θ'_i) are required. These are defined by:

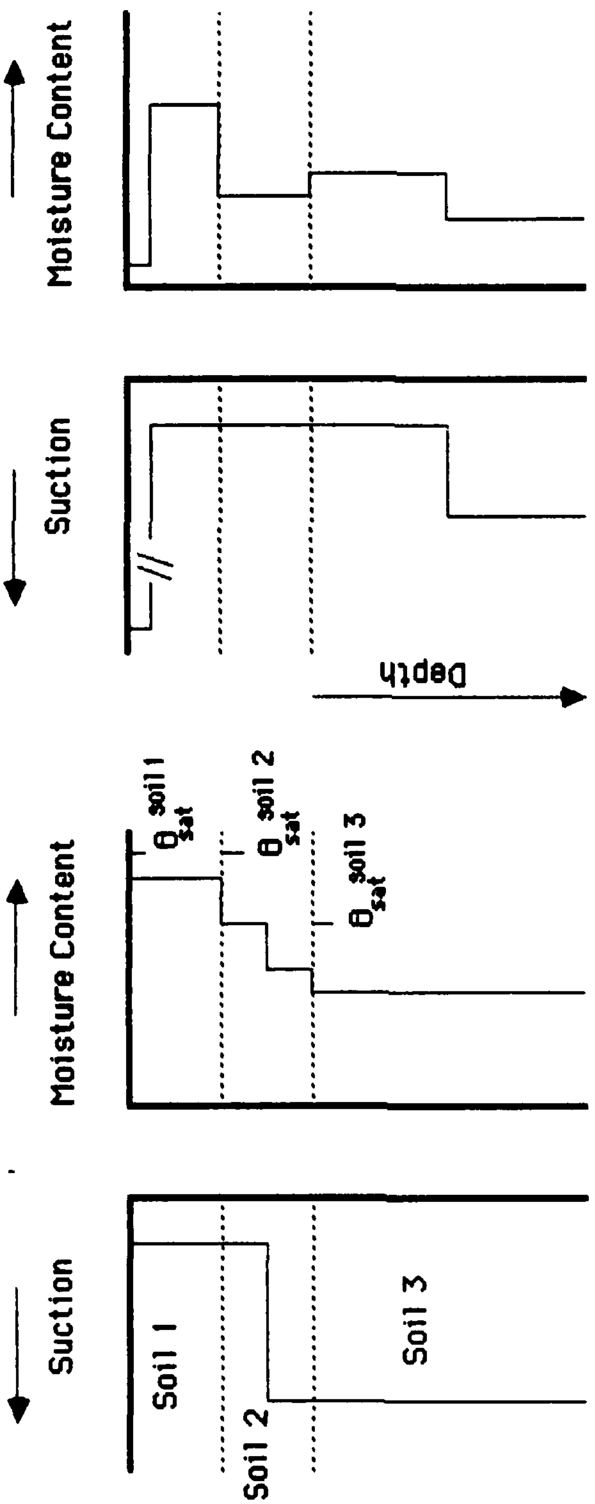
$$Q = k \left(\frac{\partial \psi}{\partial z} + 1 \right) \qquad \textbf{eq. 2.5.1}$$

where k = hydraulic conductivity

ψ = matric suction

- z = depth

FIGURE 2.8 : WETTING-FRONT SOIL MOISTURE PROFILES USING THREE DIFFERENT SOIL LAYERS



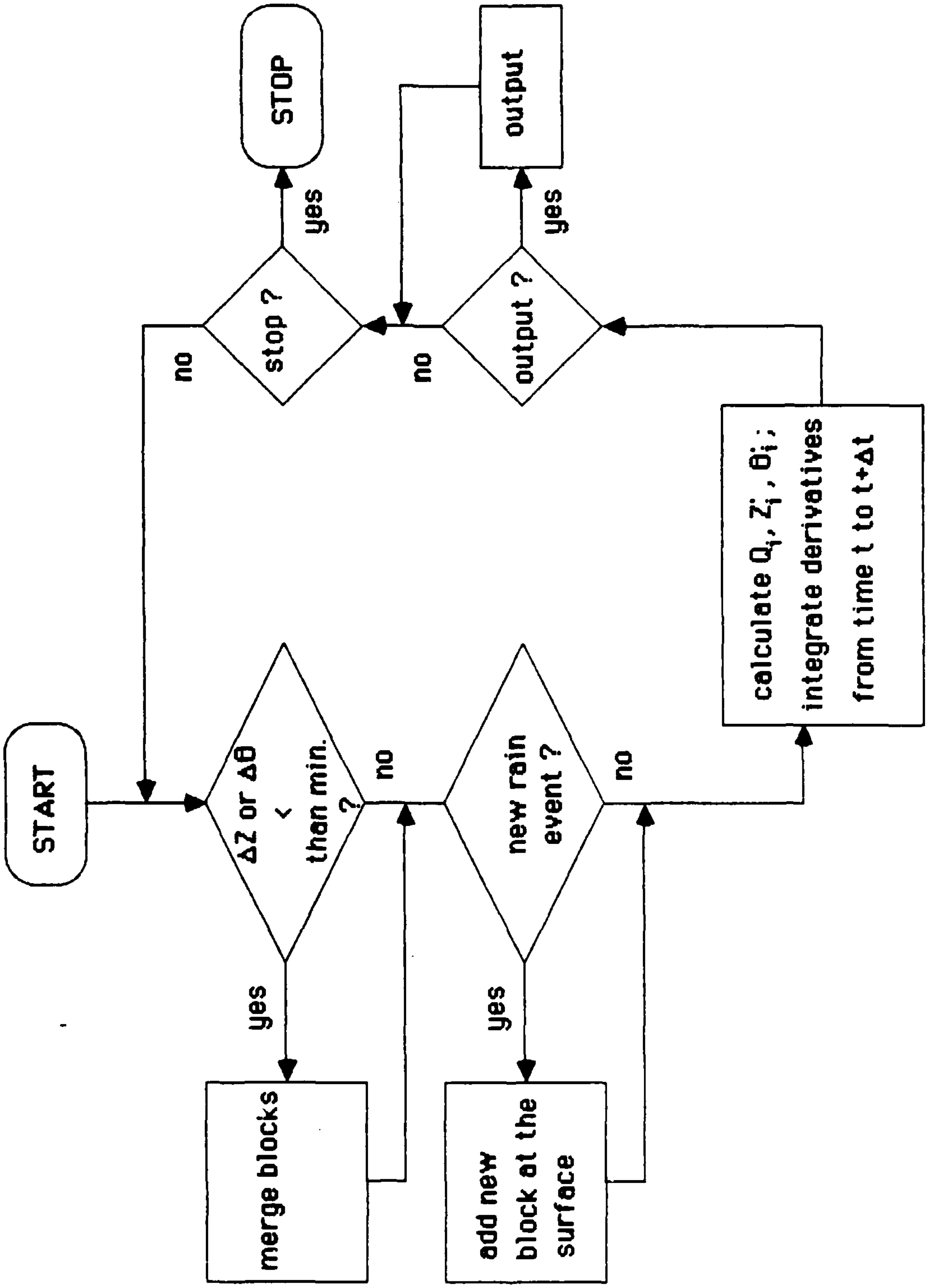
(a)

recent infiltration causes downward movement of a front

(b)

an air-dry block has formed at the surface

FIGURE 2. 9: FLOW CHART OF THE WETTING-FRONT MODEL


START
ΔZ or Δθ < than min. ?
yes
merge blocks
no
new rain event ?
yes
add new block at the surface
no
calculate Q_i, Z'_i, θ_i; integrate derivatives from time t to t+Δt
output ?
yes
output
no
stop ?
yes
STOP
no

Velocity of the wetting-front boundary is given by:

$$Z'_i = Q_i / |\Delta\theta_i| \qquad \textbf{eq. 2.5.2}$$

where $\Delta\theta_i$ = moisture difference between blocks i and i + 1

N.B. the sign of Q_i indicates the direction of movement.

The moisture content is defined by:

$$\theta'_i = (Q_i^* - Q_{i+1}^*) / \Delta Z_i \qquad \textbf{eq. 2.5.3}$$

where Q^* refers only to the fluxes that affect O_i

It was found (Clapp, 1982) that in comparison with finite difference models, daily results were within 10%, though evaporation tended to be underestimated. The subsequent revision of the evaporation routine indicated its sensitivity to changes in soil moisture at depth. The data requirements to run the wetting-front model (**table 2.4**) are modest in comparison to the previous empirical and semi-empirical models and is discussed further in **section 2.6**. The computational efficiency, i.e. 0.01 - 0.03 of the computer time required to run the finite difference type model, is an important factor in considering this model. An altered evaporation scheme would allow variable output times.

TABLE 2.4 : DATA REQUIREMENTS OF THE WETTING -FRONT MODEL

(a) Site and Storm Parameters:

precipitation
net radiation

(b) Soil Information:

boundary depths
bounday and initial conditions
saturated water content (Θ)
point of inflection (Θ)
saturated suction
saturated conductivity
soil moisture diffusivity of each layer
suction - moisture curves
conductivity - moisture curves

(c) Coefficients:

for advective effects (empirical)
for diffusivity
b, a fitted parameter that is statistically related to soil texture

2.6 Discussion and Conclusions

To fulfil the objectives of the project as set out in **section 1.2**, an empirical soil moisture prediction model such as the VB III, SMSP or the Hanl and Tries model would have been inappropriate for several reasons:-

2.6a the amount of 'difficult to evaluate' input data required (**table 2.5**),

2.6b the empirical relationships were not only site specific, but time specific, i.e. they were operational on a daily basis only (**table 2.5**),

2.6c adjusting the models to incorporate other environmental processes such as subsurface flow, frosted ground etc., would be difficult to do with any certainty of model stability.

The wetting front model (Clapp, 1982) has been designed to determine the location of the major wetting-fronts within a soil profile. This is not the information required by this project, though the use of deterministic relationships indicates that the model could be adjusted to provide the necessary information.

As defined by Smith and Meyer (1973), the most important section of the soil profile with respect to off-road trafficability, is the surface 0 - 30 cms deep layer. This indicates that a major influence in any proposed scheme might be evaporation (**section 3.4**). The evaporation submodel incorporated within Clapp's model was designed for daily calculations and was proved by Clapp to be at best 10% inaccurate. The re-design of this model and the replacement of the evaporation submodel could at best be only marginally satisfactory. The benefits of developing a scheme with the objectives set out in **section 1.2** , i.e having a structure where subroutines may easily be incorporated or exchanged, would be difficult to realize using Clapp's model as an initial basis.

TABLE 2.5 COMPARISON OF DATA REQUIREMENTS AND OUTPUT OPTIONS

	MODELS			
Information	SMSP	VB III	Hanl & Tries	Clapp
precipitation	x	x	x	x
atmos. variables	x	x	x	x
soils info.	xxx	xx	xx	x
site relations	xx	xx	xx	x
general coeffs.	x	xx	xx	x
site coeffs.	x	xx	xx	x
daily output	✓	✓	✓	✓
optional output	no	no	no	possible

KEY: x - similar amounts of data required
xx - additional information required
xxx - much more information required

Through this review of existing schemes it is now possible to re-assess and re-affirm some of the objectives set out in **section 1.2**:-

2.6d There would be much benefit if output from any newly developed scheme could be used by existing vehicle mobility models such as the CAMM (**section 2.2**) or the Ministry of Defence's DRIVEB model

2.6e The importance of having an effective evaporation submodel has been re-affirmed firstly through Smith and Meyer's work on determining that the important soil layer is 0-30 cms for off-road trafficability (**section 2.2**), and secondly, through Clapp's evaluation of the importance of evaporation on his model's performance (**section 2.5**).

2.6f The inclusion of empirical relationships and site specific relationships leads to input data requirements that are difficult and maybe impossible to evaluate for the non-expert. This re-affirms the specification of objective **1.2b**.

2.6g The difficulty of running the schemes for the non-expert indicates the need for an 'operational' and 'user friendly' system as proposed in objective **1.2c**.

3 SOIL MOISTURE MODELLING

3.1 Background

The requirements set out in **section 1.2** indicate that it will be impossible to fulfil all the requirements in one model. It is therefore proposed to develop two schemes:-

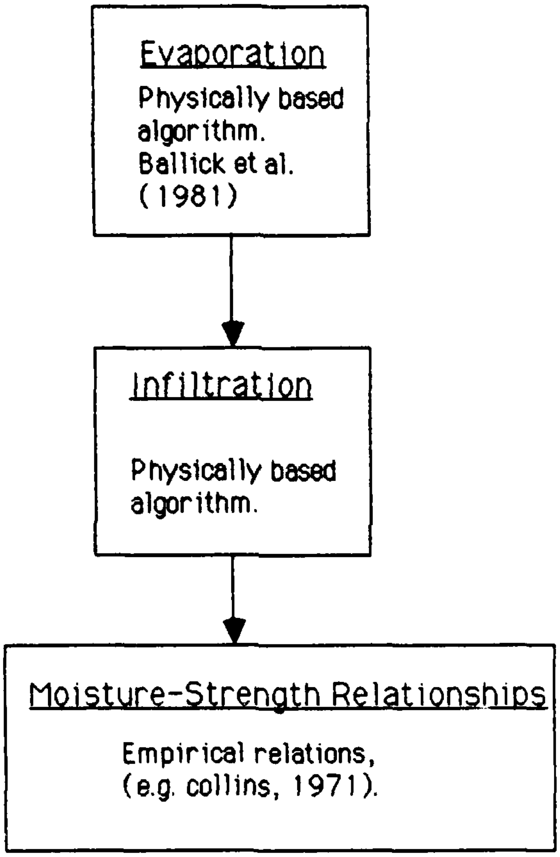
3.1a A one dimensional infiltration scheme linked with other environmental processes such as evaporation and soil strength, to be called the Bristol Soil Strength Scheme (BSSS).

3.1b A two dimensional scheme (to fulfil **requirement 1.2.1e**) coupled with soil strength. An existing scheme such as the VSAS2 will be considered for adaptation (**section 6.2**) and called VSAS2 in this project.

The BSSS is based on the basic infiltration - evaporation - soil strength scheme devised by Anderson (1983) and is shown in **figure 3.1**. Although there are existing methodologies for simulating these three processes, **section 3.2** will review the utility of various infiltration schemes of which one is technically described in **section 3.5** for use within the BSSS. Inherent within the infiltration methodologies is the calculation of hydraulic conductivity (**section 3.3**) and the matric potential (**section 3.4**). The appropriate technical description of the chosen methods to be used in the BSSS are give in **section 3.5**.

The specification for the evaporation scheme are described in **section 3.7**. A technical description of the method is given though the scheme requires some further development which is described in **section 4.2**.

FIGURE 3.1 INITIAL ANDERSON MODEL (1983)


Evaporation
Physically based algorithm. Ballick et al. (1981)
Infiltration
Physically based algorithm.
Moisture-Strength Relationships
Empirical relations, (e.g. collins, 1971).

The soil strength scheme is reviewed in **section 4.1** and a completely new scheme is developed.

The VSAS2 scheme (Bernier, 1982; Whitelaw, 1988) simulates the two dimensional flow and convergence of water within a slope. It is a more complicated model and as such requires much more input from the user and has significantly different computational requirements from the BSSS (**section 10**). The infiltration scheme used in the VSAS2 is technically described in **section 3.6** along with the methodologies used to calculate hydraulic conductivity and matric potential. There is no evaporation scheme incorporated into the VSAS2 and this restriction is further discussed in **section 8.2**.

3.2 Review of suitable One Dimensional Infiltration and Redistribution Schemes

The modelling of infiltration and redistribution of water within a soil profile can be divided into three components:-

(a) The actual modelling of the flow of water.
(b) The calculation of hydraulic conductivity (**section 3.3**).
(c) The calculation of matric potential(**section 3.4**).

This section will discuss the flow of soil water within the soil body while assuming that a satisfactory method has been used to determine the hydraulic conductivity and matric potential.

Many relationships describing infiltration as a function of time or the amount of water infiltrated into the soil have been devised. Some are entirely empirically and others theoretically based. It is this second category of relationships, i.e. the theoretically based equations that we will review in the effort to fulfil **requirement 3.5.2a** specified in **section 3.5** for inclusion in the BSSS.

The extent of the relationships reviewed is also limited to those that are well established (**requirement 3.5.2h**). Hence, those relationships devised by Philip (1957), Green-Ampt (1911) and the Richard's equation (Swartzendruber, 1969) will be considered. Established relationships such as the Hortan (1940) and Holtan (1961) equations are empirical in nature, requiring calibration to select the correct qualitative shape.

3.2.1 Philip's Equation

Philip (1957) proposed that infiltration is described by:

$$\frac{\partial q}{\partial t} = s/2t^{1/2} + c_a$$

where $\partial q/\partial t$ = volume of water entering a unit soil surface area per unit time

s = sorptivity = $(2MK_sS_f)^{1/2}$ (Youngs, 1968)

c_a = characterizing constant = $2K_s/3$ (Youngs, 1968)

S_f = effective matric suction at the wetting front

M = pore space available for water, i.e. $\theta_s - \theta_i$

K = hydraulic conductivity

Philip's equation originated from the first two terms of his series solution for infiltration from a ponded surface into a deep homogeneous soil. As such, the relationship is inherently restrictive as to the form of water application, i.e., the relationship implies surface ponding from the start of the simulation period. Philip's methods for defining c_a were shown by Youngs (1968) to be physically inconsistent at large run times and hence limited the application to short run times.

3.2.2 Green-Ampt Equation

Green and Ampt (1911) utilized Darcy's equation to simulate the progress of a 'slug' of water, defined by a wetting-front, through a soil profile. This is the methodology used by Clapp (1982) in his wetting-front model reviewed in **section 2.5**. It is described by:

$$f = K_s + K_s M S_f / F \qquad \textbf{eq. 3.2.2a}$$

where f = infiltrability

K_s = hydraulic conductivity of the transmission zone

M = pore space available for water, i.e. $\theta_s - \theta_i$

S_f = effective matric suction at the wetting front

F = cumulative infiltration = ML_f

L_f = distance from the surface to the wetting front

The derivation of this equation assumes a ponded surface so that the infiltration is at all times equal to the infiltration capacity. Further development of the Green-Ampt equation carried out by Bouwer (1966) showed that the hydraulic conductivity should be less than the saturated hydraulic conductivity because of entrapped air. The evaluation of the wetting front suction is also difficult to evaluate and consequently several methodologies were formulated to aid in the evaluation of these parameters. The work carried out by Brakensiek (1977) using the prediction methods by Brooks and Corey (1964), is often used to calculate the wetting front suction. The more accepted method of determining the

Green-Ampt parameters (Haan et al., 1982) is given by Brakensiek and Onstad (1977) as the fitting of infiltrometer data.

Further work carried out by Mein and Larson (1973) developed the scheme for conditions where the infiltration rate was not equal to the infiltration capacity. Using **equation 3.2.2a**, S_f is substituted by S_{av}, the average suction at the wetting front and F by F_p, where F_p is the cumulative infiltration at the time of ponding ($t=t_p$) . **Equation 3.2.2a** is solved for F_p as follows:

$$F_p = \frac{S_{av}M}{R/K_s - 1} \qquad \textbf{eq. 3.2.2b}$$

since $f = R$ prior to surface ponding, $F_p = Rt_p$

where t_p = time of surface ponding

For a steady rainfall rate the infiltration rate is therefore expressed by:

$f = R$ when $t < t_p$

$f = K_s + K_s S_{av} M/F_p$ when $t > t_p$

If $R < K$ surface ponding will not occur. The application of the Green-Ampt equation to unsteady rainfall conditions has been examined (Reeves and

Miller, 1975; James and Larson, 1976; Chu, 1978) and Haan et al. (1982) concluded the following:

(i) The Green-Ampt equation will give a good approximation of infiltration during unsteady rainfall resulting in an extension of the wetting front.

(ii) Over long periods of low intensity or no rainfall the wetted profile will redistribute.

From these results it is possible to say that the utility of the Green-Ampt equation is limited to periods of a minimum intensity rainfall. The equations will not be reliable over longer drier periods.

3.2.3 Richard's Equation

Richard's equation is the basis of infiltration theory (Hillel, 1982) which combines Darcy's equation with the continuity equation to solve the general flow equation of water in soil. The use of the Richard's equation is widespread and well established. It is the methodology used in the prototype (Anderson, 1983) of the proposed scheme. As the other methodologies do not indicate any major advantages over the Richard's equation, and infact are less suited to the purpose, the Richard's equation will be used to model one dimensional infiltration within the BSSS. A technical description of the one dimensional version is given in **section 3.5.3** and the limitations of the scheme are discussed in **section 8.1**.

The two dimensional version is used within the VSAS2 and is technically described in **section 3.6.3**. Both versions of Richard's equation require the calculation of hydraulic conductivity and matric suction as related to soil moisture content and therefore the following section will review the methodologies available for these calculations.

3.3 Review of Schemes to Calculate Hydraulic Conductivity

To incorporate principles of soil physics such as the infiltration equations of Green and Ampt (1911), Philip (1957) or Darcy (1856) etc., the conductivity function, i.e., the relationship between unsaturated hydraulic conductivity (k) and soil moisture content (O), must be known.

Various equations have been developed for the relation of conductivity to suction or wetness. Some of these are given in **table 3.1**, where it can be seen that they are all either entirely or semi-empirical based. It can be seen that there is a marked similarity in the first six relationships where initial development by Childs and Collis-George (1950) and Marshall (1958) has been compared with measured hydraulic conductivities. The evaluation of the exponential constant has been 'fitted' by each author from their results.

The original development of the relationships of Childs and Collis-George (1950) was based on the following assumptions (Childs, 1969):-

(a) that soil water flow is controlled by the smaller pore in a sequence,
(b) that only pores in a direct sequence contribute to the total hydraulic conductivity, and
(c) that the pores in the soil medium fit together randomly.

Therefore, the hydraulic conductivity for a given soil water content is the sum of the contributions to conductivity of each pore class between radius zero and the radius of the largest water-filled pores.

There have been many comparisons of calculated and measured hydraulic conductivity (e.g. Nielsen et al., 1960; Jackson et al., 1965; Green and

TABLE 3.1 SOME METHODS OF CALCULATING HYDRAULIC CONDUCTIVITY

RELATIONSHIP	TYPE	AUTHOR
$k(\theta) = k_s(\theta / \theta_s)^{2b+3}$	Partially empirical & partially theoretical	Jackson (1972)
$k(\theta) = k_s(\theta / \theta_s)^{2b+2}$	Partially empirical & partially theoretical	Campbell (1974)
$k(\theta) = k_s(\theta / \theta_s)^{1.5b+3}$	Partially empirical & partially theoretical	Gosh (1977)
$k(\psi) = k_s(\psi_e / \psi)^{2+2/b}$	Partially empirical & partially theoretical	Campbell (1974)
$k(\psi) = k_s(\psi_e / \psi)^{2+3/b}$	Partially empirical & partially theoretical	Jackson (1972)
$\log_{10} k = b\,\lvert\psi\rvert^{1/2} + a$	Empirical	Gardner (1958)
$k_i = k_s(\theta_i / \theta_s)^c \dfrac{\sum_{j=i}^{m}[(2j+1-2i)\psi_j^{-2}]}{\sum_{j=1}^{m}[(2j-1)\psi_j^{-2}]}$	mainly theoretical with matching factor	Millington and Quirk (1959); Jackson (1972), Campbell (1974) Marshall (1958)

where k_s = saturated hydraulic conductivity

ψ = matric suction

a & b = empirical constant

c = pore interaction term

θ = soil water content

i & j = consecutive soil moisture intervals

Corey, 1971; Kunze et al. 1968). Jackson (1972) concluded from these investigations that the Marshall and the Millington Quirk methods were both the easiest to use and provided reasonable results when an appropriate matching factor was used. Jackson (1972) showed that when a matching factor is used that these two methods are only different in the pore interaction term.

For practical application the difference between the Millington Quirk and the Marshall methods is in the 'c' and 'n' terms:-

Term	**Millington Quirk**	**Marshall**
c	pore interaction term = 4/3 (Millington Quirk, 1959) = 1 (Jackson, 1972; Kunze et al. 1968)	pore interaction term = 2 (Marshall, 1958) = 0 (Jackson, 1972)
n	total number of water content increments from zero to the saturated water content	number of water content increments from zero to the water content in question

Both Jackson (1972) and Kunze et al. (1968) determined that a pore interaction value of one in the Millington Quirk method, adequately predicted the measured conductivities and would be applicable to calculating the hydraulic conductivity - soil moisture content relation for field soils.

3.4 Review of Schemes to Calculate Matric Potential

The infiltration relationships discussed in **section 3.2** and consequently the methods used to calculate hydraulic conductivity in **section 3.3** require a methodology to calculate the moisture characteristic, i.e., the relationship between suction (ψ) and the soil moisture content.

Several empirical relations have been developed to calculate matric potential or soil water potential and are shown in **table 3.2**. The most commonly used relationship is the power function of Campbell (1974) inwhich both the suction at saturation (ψ_s) and the exponent (b) are empirical and have to estimated. Clapp and Hornberger (1978) used data collected by Holtan et al. (1968) to calculate ψ_s and 'b' through a linear regression. The results were somewhat problematic in that:-

(a) results from rocky soils were excluded because they were too erratic,
(b) results from soils where the calculation of θ/θ_s exceeded unity at 0.1 bar suction were excluded, and
(c) results which gave 'b' values greater than 25 were also excluded, as exponents of this magnitude were considered anomalous.

From the 1446 soils eventually used, Clapp and Hornberger compiled values for U_s and 'b' for the main USDA soil textural classes and is shown in **table 3.3**.

Soil water retention at selected matric potentials have been correlated with the physical soil properties such as particle size, organic matter and soil bulk density (Gupta and Larson, 1979; Rawls, Brakensiek and Saxon,

TABLE 3.2 SOME METHODS OF CALCULATING SOIL WATER POTENTIAL

RELATIONSHIP	TYPE	AUTHOR
$\psi(\theta) = \psi_s(\theta/\theta_s)^{-b}$	Empirical	Campbell (1974)
$\psi = \psi_{cr} e^{a(1-Se)}$	Empirical	Farrell and Larson (1972)
For each given ψ:- $\theta = a_i + b_i(\%sand) + c_i(\%clay) + d_i(\%om) + e_i(sbd)$	Empirical	Rawls, Brakensiek and Saxon (1982) Mien and Larson (1971)

where k_s = saturated hydraulic conductivity

Se = effective saturation

ψ = matric suction

ψ_{cr} = matric suction at which $\partial\theta/\partial\psi > 0$

$a_i - e_i$ = regression coefficients

a = empirical constant

θ = soil water content

i & j = consecutive soil moisture intervals

om = organic matter

sbd = soil bulk density

TABLE 3.3 REPRESENTATIVE VALUES FOR HYDRAULIC PARAMETERS

(Clapp and Hornberger, 1978)

Soil Texture Class	b	ψ_s (cm)
Sand	4.05	12.1
Loamy sand	4.38	9.0
Sandy loam	4.90	21.8
Silt loam	5.30	78.6
Loam	5.39	47.8
Sandy clay loam	7.12	29.9
Silty clay loam	7.75	35.6
Clay loam	8.52	63.0
Sandy clay	10.40	15.3
Silty clay	10.40	49.0
Clay	11.40	40.5

1982). Rawls, Brakensiek and Saxon (1982) assembled soil information for 1323 soils with about 5,350 horizons from 26 sources of data. Their results produced reasonably good correlation coefficients (0.8 - 0.95) when compared to the Gupta and Larson equations (1979). Rawls, Brakensiek and Saxon (1982) developed three levels of linear regressions relating soil water retention at specific potentials to:-

(a) % sand, % clay, % silt, % organic matter and soil bulk density;

(b) % sand, % silt, % clay, % organic matter, soil bulk density and 15 bar water retention;

(c) % sand, % silt, % clay, % organic matter, soil bulk density, 0.33 and 15 bar water retention.

Further analyses by Brakensiek and Rawls (1983) produced a linear regression based on :-

(d) % sand, % clay, % organic matter and soil bulk density.

This provides a range of equations the utility of which will be dependent on the availability of soil information. The research of Brakensiek and Rawls (1983) also developed methodologies of estimating these soil characteristics for the USDA soil textural classes and if further discussed in **section** 4.3.

3.5 One Dimensional Soil Water Movement

3.5.1 Objectives

As discussed in **section 3.1**, an acceptable method for simulating the the flow of water through a soil profile is a necessary attribute of the proposed BSSS. The following sections refine the system requirements set out in **section 1.2**, and in the light of these requirements review schemes that might fulfil as many of those requirements as possible.

3.5.2 Detailed Requirements

Requirement 1.2.1d specifies a 'physically based system to predict soil moisture'. This term 'physically based' means 'deterministic relationships describe the physical processes of water movement operating within the soil profile'. Hence, the specific requirements may be defined as follows:-

3.5.2a The infiltration and redistribution of water within a one dimensional soil profile should be simulated by a physically based scheme, i.e., the behaviour of the scheme is represented by a set of mathematical equations along with logical statements expressing relationships between variables and parameters.

3.5.2b The scheme should not require calibration, i.e., calibration is where input to a function is neither directly measurable or logically calculated and must be evaluated through the methodology of "fitting".

3.5.2c The scheme adopted should have input data that is easily acquired by the trained but non-expert user.

3.5.2d The system can provide any 'difficult' to acquire input data from a simple input from the user, i.e., data banks are permissible.

3.5.2e The scheme should have variable temporal resolution.

3.5.2f The scheme should have the potential for integration with other schemes to predict environmental factors such as evaporation and soil strength.

3.5.2g The design and code of the system should be fully described and documented. The description should include a technical description, the input data required and the restrictions of application.

Potential schemes for calculating infiltration, hydraulic conductivity and matric potential were reviewed in **sections 3.2, 3.3** and **3.4** respectively. The scheme that will fulfil as many of the above requirements as possible will be chosen from those reviewed, but, it should be remembered that the scheme eventually adopted will probably be a compromise and therefore the different requirements will take on different priorities. Another requirement should therefore be specified:-

3.5.2h The scheme adopted should be well established and tested to eliminate the need for an excess of time to be spent on this one facet of the project.

This last requirement tries to bring into perspective this section has within the main objective specified in **section 1.2**, i.e., the main purpose of this project is '*to develop an operational system to predict off-road trafficability on a km^2 scale for application to route management problems*'.

3.5.3 Technical Description of Chosen Methods

3.5.3.1 : One Dimension Infiltration (According to Darcy, (1856; Hubbert, M.K., 1956) and Richards, (Swartzendruber, D., 1969))

The movement of water between cells (**figure 3.2**) is defined according to Darcy's Law:-

$$\frac{\partial \Theta}{\partial t} = \frac{\partial}{\partial z}\left(k(\Theta)\frac{\partial \Psi}{\partial z}(\Theta)\right) - \frac{\partial k}{\partial z}(\Theta) \qquad \textbf{eq. 3.5.3.1a}$$

where Θ = soil water content
t = time
z = distance
$k(\Theta)$ = hydraulic conductivity at soil water content Θ
Ψ = matric potential

Equation 3.5.3.1a is solved through the following equations:-

Richard's equation for flow

$$q = k(\Theta)\,\Delta h \qquad \textbf{eq. 3.5.3.1b}$$

where q = apparent water velocity
$k(\Theta)$ = hydraulic conductivity at soil water content 0
Δh = hydraulic gradient

Continuity states that:-

$$\frac{\partial \Theta}{\partial t} = \Delta q \qquad \textbf{eq. 3.5.3.1c}$$

where Θ = soil water content

t = time

Δq = flow gradient

Substitute **eq. 3.5.3.1c** into **eq 3.5.3.1b**:-

$$\frac{\partial\Theta}{\partial t} = \Delta\left(k(\Theta)\Delta h\right) \qquad \textbf{eq. 3.5.3.1d}$$

Set

$$\Delta h = \Psi - z \qquad \textbf{eq. 3.5.3.1e}$$

where $\Psi = \Psi(\Theta)$ = matric potential at soil water content Θ, and may be either a suction or a pressure

z = gravitational head (or depth)

Substitute **eq. 3.5.3.1e** into **eq. 3.5.3.1d**:-

$$\frac{\partial\Theta}{\partial t} = \Delta\left(k(\Theta)\Delta(\Psi - z)\right) \qquad \textbf{eq. 3.5.3.1f}$$

Expand **eq. 3.5.3.1f** in Z direction:-

$$\frac{\partial\Theta}{\partial t} = \frac{\partial}{\partial Z}\left[\, k(\Theta)\left(\frac{\partial\Psi}{\partial z} - \frac{\partial z}{\partial z}\right)\right]$$

$$\Rightarrow \quad \frac{\partial\theta}{\partial t} = \frac{\partial}{\partial Z}\left[\frac{k(\theta)\partial\psi(\theta)}{\partial z} - k(\theta) \right]$$

$$\Rightarrow \quad \frac{\partial\theta}{\partial t} = \frac{\partial}{\partial z}\left[\frac{k(\theta)\partial\psi(\theta)}{\partial z} \right] - \frac{\partial k(\theta)}{\partial z} \qquad \textbf{eq 3.5.3.1g}$$

(One dimensional flow)

3.5.3.2 Calculation of Hydraulic Conductivity in the BSSS

Calculate the hydraulic conductivity according to the relationship established by Millington and Quirk (1959) and developed by Jackson (1972) and Campbell (1974) :-

$$k_i = \frac{k_s \ (\theta_i / \theta_s)^c \ \sum_{j=i}^{m} [(2j+1-2i)\psi_j^{-2}]}{\sum_{j=1}^{m} [(2j-1)\psi_j^{-2}]} \qquad \textbf{eq. 3.5.3.2a}$$

at equal moisture intervals (Millington and Quirk).

where k_s = hydraulic conductivity

θ_s = saturated soil moisture content

c = a pore interaction constant = 1

m = number of equal sized moisture intervals

3.5.3.3 Calculation of Matric Potential in the BSSS

Calculate soil bulk density for each different soil type according to Rawls, Brakensiek and Saxon, (1982):-

$$\text{soil bulk density} = (100/(om\%/0.224)+((100-om\%)/mbd)))$$

eq. 3.5.3.3a

where om% = percentage organic matter content

mbd = mineral bulk density

Calculate the 10 point **θ-Ψ curve** according to Rawls, Brakensiek and Saxon, (1982):-

For each given Ψ

$$\theta = a_i + b_i\ (\%\ sand) + c_i\ (\%\ clay) + d_i\ (\%\ om) + e_i\ (sbd)$$

eq. 3.5.3.3b

where θ = soil moisture content

Ψ = matric suction

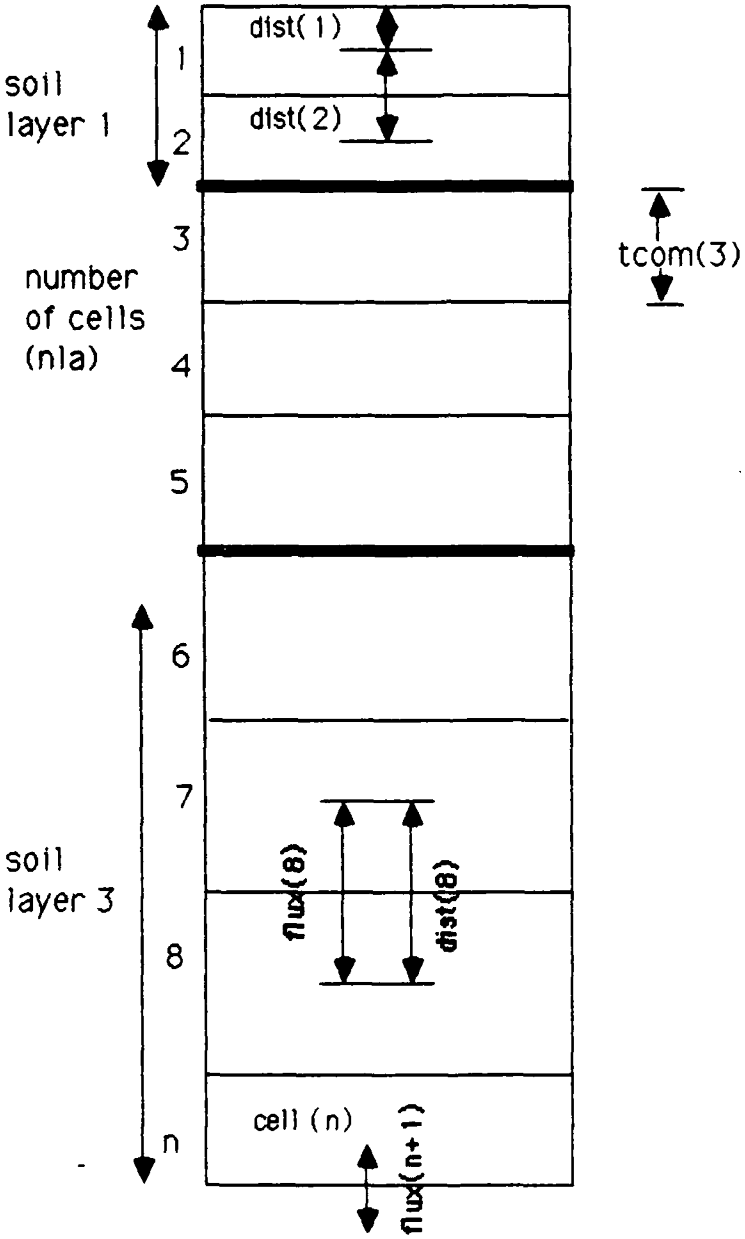
om = organic matter

sbd = soil bulk density

coefficients a-e are abstracted from the following table for values of ψ:

Matric					
Potential	a	b	c	d	e
(metres)					
- 2.04	0.4180	-0.0021	0.0035	0.0232	-0.0859
- 3.36	0.3486	-0.0018	0.0039	0.0228	-0.0738
- 6.11	0.2819	-0.0014	0.0042	0.0216	-0.0612
- 10.19	0.2352	-0.0012	0.0043	0.0202	-0.0517
- 20.38	0.1837	-0.0009	0.0044	0.0181	-0.0407
- 40.76	0.1426	-0.0007	0.0045	0.0160	-0.0315
- 71.33	0.1155	-0.0005	0.0045	0.0143	-0.0253
-101.90	0.1005	-0.0004	0.0044	0.0133	-0.0218
-152.85	0.0854	-0.0004	0.0122	0.0122	-0.0182

FIGURE 3.2 : MOVEMENT OF WATER BETWEEN CELLS



3.6 Two Dimensional Soil Water Movement

3.6.1 Objectives

The general requirements set out in **section 1.2.1** highlight the necessity of developing a facility capable of handling a steep area that may be highly vegetated The solution to this requirement has been to adapt an existing scheme (VSAS2) for application to soil strength calculations (**section 6.2**). The following section sets out the requirements which the VSAS2 incorporates within it which are of use for adaptation. The soil physics methodologies used in the VSAS2 are then technically described (**section 3.6.3**).

3.6.2 Detailed Requirements

The requirements which apply to development of the one dimensional infiltration scheme, i.e. physically based, also apply to the development of the two dimensional scheme. **Requirement 1.2.1e** has been re-defined as follows:-

3.6.2a The infiltration and redistribution of water within a sloping segment should be simulated by a physically based two dimensional scheme.

3.6.2b The scheme should not require calibration.

3.6.2c The scheme should have variable temporal resolution.

3.6.2d The scheme should have the potential for integration with other schemes to predict soil strength.

3.6.3e The design and code of the system should be fully described and documented. The description should include a technical description, the input data required and the restrictions of application.

3.6.3f The schemes adopted should be well established and tested.

Potential schemes for calculating infiltration, hydraulic conductivity and matric potential were reviewed in **sections 3.2, 3.3** and **3.4** respectively. The methods used in the VSAS2 for calculating two dimensional infiltration and redistribution, hydraulic conductivity and matric potential are technically described in the following section. **Section 6.2** describes the design and adaptation of the VSAS2 for application to off-road trafficability predictions.

3.6.3 Technical Description of Chosen Methods

3.6.3.1: Two Dimensional Infiltration (According to Darcy, (1856; Hubbert, 1956) and Richard's, (Swartzendruber, 1969))

RICHARD'S EQUATION OF FLOW

$$q = k(\theta)\ \Delta h \qquad \textbf{eq. 3.6.3.1a}$$

where q = apparent water velocity

$k(\theta)$ = hydraulic conductivity at soil water content θ

Δh = hydraulic gradient

CONTINUITY STATES THAT:-

$$\frac{\partial\theta}{\partial t} = \Delta q \qquad \textbf{eq. 3.6.3.1b}$$

where θ = soil water content

t = time

Δq = flow gradient

Substitute **eq. 3.6.3.1b** into **eq 3.6.3.1a**:-

$$\frac{\partial\theta}{\partial t} = \Delta\left(k(\theta)\Delta h\right) \qquad \textbf{eq. 3.6.3.1c}$$

Set

$$\Delta h = \psi + z \qquad \textbf{eq. 3.6.3.1d}$$

where $\psi = \psi(\theta)$ = matric potential at soil water content θ, and may be either a suction or a pressure

z = gravitational head

Substitute **eq. 3.6.3.1d** into **eq. 3.6.3.1c**:-

$$\frac{\partial\theta}{\partial t} = \Delta\left(k(\theta)\Delta(\psi+z)\right) \qquad \textbf{eq. 3.6.3.1e}$$

Expand **eq. 3.6.3.1e** in x and z directions

$$\frac{\partial\theta}{\partial t} = \frac{\partial}{\partial x}\left[\, k(\theta)_x \left(\frac{\partial\psi}{\partial x} + \frac{\partial z}{\partial x}\right) \right] + \frac{\partial}{\partial z}\left[\, k(\theta)_z \left(\frac{\partial\psi}{\partial z} + \frac{\partial z}{\partial z}\right) \right]$$

eq. 3.6.3.1f

From Bernier (1982), x represents slope, i.e. $x \cos \beta$

where β = slope angle

$$x^{*} = x \cos \beta \qquad \textbf{eq. 3.6.3.1g}$$

Substitute **eq. 3.6.3.1g** into eq. **3.6.3.1f**

$$\frac{\partial \theta}{\partial t} = \cos^2 \beta \, \frac{\partial}{\partial x^*}\left[k(\theta)_{x^*} \left(\frac{\partial \psi}{\partial x^*} + \frac{\partial z}{\partial x^*}\right) \right] + \frac{\partial}{\partial z}\left[k(\theta)_z \left(\frac{\partial \psi}{\partial z} + 1\right) \right]$$

eq. 3.6.3.1h

(Two dimensional flow)

Expand **eq. 3.6.3.1e** in Z direction:-

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial Z}\left[k(\theta) \left(\frac{\partial \psi}{\partial z} + \frac{\partial z}{\partial z}\right) \right]$$

$$\Rightarrow \quad \frac{\partial \theta}{\partial t} = \frac{\partial}{\partial Z}\left[\frac{k(\theta)\partial \psi(\theta)}{\partial z} + k(\theta) \right]$$

$$\Rightarrow \quad \frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z}\left[\frac{k(\theta)\partial \psi(\theta)}{\partial z} \right] + \frac{\partial k(\theta)}{\partial z}$$

eq 3.6.3.1i

(One dimensional flow)

3.6.3.2 : Campbell's Method of Calculating Hydraulic Conductivity

$$k(\theta) = k_{sat} \left(\frac{\theta}{\theta_{sat}}\right)^{2b+2} \qquad \textbf{eq. 3.6.3.2a}$$

where $k(\theta)$ = hydraulic conductivity at soil water content 0

θ = soil water content

θ_{sat} = saturated soil water content

k_{sat} = saturated hydraulic conductivity

b = empirical coefficient (**table 3.3** gives some values for the different soil textural classes)

Average Hydraulic Conductivity Between Two Different Soil Elements (Childs, 1969)

$$\bar{K} = \frac{D_1 + D_2}{\frac{D_1}{k(\theta)_1} + \frac{D_2}{k(\theta)_2}} \qquad \textbf{eq. 3.6.3.2b}$$

where k = average hydraulic conductivity

$k(\theta)_1$ = hydraulic conductivity of element 1

$k(\theta)_2$ = hydraulic conductivity of element 2

D_1 = thickness of element 1

D_2 = thickness of element 2

3.6.3.3 Campbell's Method of Calculating Matric Potential

$$U(\Theta) = \Psi_e \left(\frac{\Theta}{\Theta_{sat}}\right)^{-b}$$ **eq. 3.6.3.3a**

where $\Psi(\Theta)$ = matric potential at soil water content Θ

Θ = soil water content

Θ_{sat} = saturated soil water content

Ψ_e = air entry potential

b = empirical coefficient (**table 3.3** gives some values for the different soil textural classes)

3.7 Calculation of Evaporation

3.7.1 Objectives

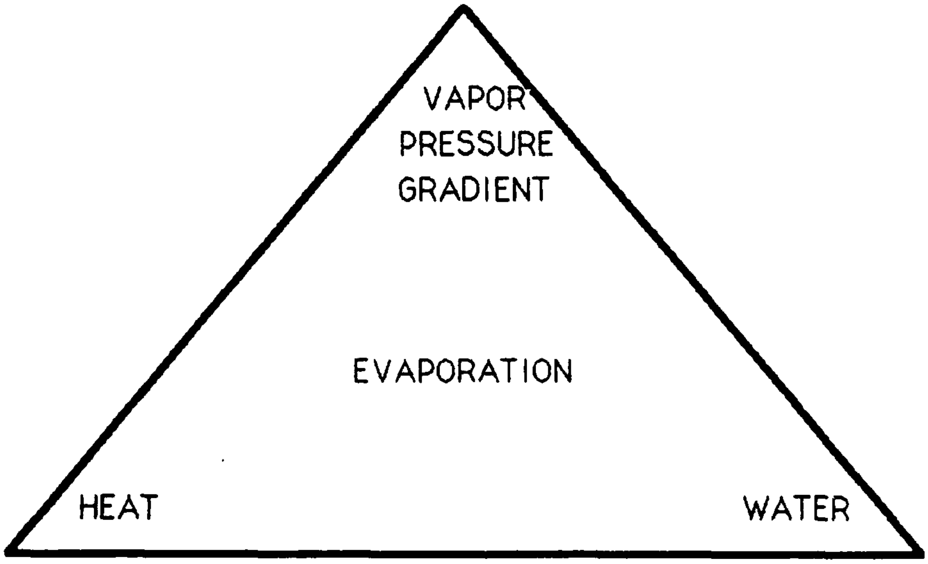
The importance of the surface layer (0-15cms) in trafficability calculations was determined by Smith and Meyer (1973) and the discussion in **section 2** highlighted the importance of evaporation on this layer. The temporal resolution of the proposed BSSS indicates the need for an evaporation methodology which also operates on a variable temporal scale.

Evaporation from a bare soil surface occurs when affected by radiation and wind. Evaporation from plants (transpiration) is affected by the evaporative demand of the surrounding climate. By limiting the use of the BSSS to temperate regions, the dominant process is considered to be evaporation. Hence, at this first attempt, a satisfactory method of estimating the variable nature of evaporation is considered as a prime objective. Some effect of vegetation is included in the choice of albedo (**section 4.2**).

For evaporation to take place from a surface three physical conditions must be met. These are represented by the 'evaporation triangle' shown in **figure 3.3** . There must be an adequate supply of heat to satisfy the latent heat requirement of water. The source of this heat can be from the atmosphere in the form of radiated or advected energy, or from the body itself. The most probable and dominant source of energy in the temperate latitudes will be that from the atmosphere.

For evaporation to take place there must also be a vapor pressure gradient between the soil body and the atmosphere, i.e., the vapor pressure in the

FIGURE 3.3 EVAPORATION TRIANGLE



For evaporation to take place there must be heat, a vapor pressure gradient and water. If one of these conditions is not met there will be no evaporation.

atmosphere over the body must be lower than the vapor pressure at the surface of the body.

The final requirement for evaporation to take place, as shown in **figure 3.3**, is that of an adequate supply of water for evaporation. The calculation of the amount of water in the profile at any given point in time has been covered in **section 3.5** and therefore it is the first two requirements of the evaporation triangle that will be examined to provide a methodology for estimating evaporation.

The supply of energy and the removal of vapor are mainly characteristics of the surrounding atmosphere. The rate at which evaporation can take place will be influenced by meteorological factors such as radiation, air temperature, humidity and wind velocity. These environmental conditions may remain constant or fluctuate over the required time period, i.e. diurnal, or also over a longer time period, i.e. seasonally or annually.

Any scheme adopted to estimate evaporation must be able to work at any time of the year and because of the temporal resolution required, i.e. output at intervals of 0.25 - several hours, the diurnal variation should be incorporated. This indicates that a non-isothermal model of evaporation estimation is the type of model that would best suit the model needs.

3.7.2 Detailed Requirements

The section above discussed the general objectives used to determine the type of evaporation model that would best suit the BSSS. A further refinement of those objective and the requirements set out in **section 1.2** is now necessary to define a proposed scheme.-

(a) The system should be as physically based as possible.

(b) The scheme should require only meteorological information that is readily available from a standard meteorological recording station, i.e. humidity, air and ground temperatures, windspeed, cloud information and atmospheric pressure.

(c) The scheme must be able to operate at any time of the year, i.e. information about the solar zenith angle etc. can be provided.

(d) The scheme must display the diurnal fluctuation in the evaporative ability of the environment.

(e) The scheme should allow for some influence of topography on the evaporation estimation, i.e., the orientation and angle of slope.

These requirements are quite specific in the type of scheme required to estimate evaporation and consequently eliminate almost all of the most commonly used empirical or isothermal evaporation routines.

3.7.3 Review of Suitable Schemes

The requirements specified above specify a system that is highly complex and dynamic in nature. An energy balance approach has been taken by van Bavel and Hillel (1975, 1977); Khale (1977) and Balick et al. (1981).

The scheme developed by van Bavel and Hillel (1977) uses radiation air temperature, albedo, humidity and wind speed to predict sensible heat, latent heat and the pattern of soil moisture and soil temperature in the profile. The energy balance part of the scheme calculates incoming and outgoing long-wave radiation and then solves for evaporation.

The Balick et al. scheme uses the energy balance equation developed by Khale (1977) which requires radiation, air and ground temperatures, humidity, albedo, cloud information, humidity, atmospheric pressure and solar zenith angle as inputs. Balick et al. also require site information to assess the influence of slope orientation and steepness on evaporation estimation.

The Balick et al. scheme was part of the original Anderson (1983) model and with correction of the albedo handling (**section 4.2**) this is the model which fulfills the requirements set out above. The full technical description of the scheme is given in **section 3.7.4**.

3.7.4 Technical Description of the Non-Isothermal Evaporation Routine

Calculate evaporation at any time of day and year according to Weisner's (1970) method :-

$$\text{evaporation} = \frac{S - I - G - H}{Lx} \qquad \textbf{eq. 3.7.4a}$$

where S = incoming radiation to a surface of albedo a_g

I = thermal infra-red radiation

G = heat flux from the ground

Lx = latent heat exchange

H = sensible heat

Calculate S according to:-

$$S = (1-a_g)\,[1-A(u^*,z)](0.349)S_0 \cos z \qquad \textbf{eq. 3.7.4b}$$

$$+ (1-a_g)\,[(1-a_0)/(1-a_0\underline{a}_g)](0.651)S_0 \cos z$$

where S = incoming radiation at the ground with no cloud cover

a_g = surface albedo

a_g = average ground albedo

a_0 = atmospheric albedo for Raleigh scattering , equal to

$$0.085 - 0.247 \log_{10} [(p_s/p_0)\cos z]$$

p_s = surface pressure

p_o = 1000 mb

z = zenith angle of the sun as a function of the time of day and year

$A(u^*,z)$ = Mugge - Moller absorption function, equal to $0.271(u^* \sec z)^{0.803}$

u^* = effective water vapour content of the atmosphere

S_0 = solar radiation incident on top of the atmosphere

$(0.349)S_0$ = amount of solar radiation of wavelength > 0.9 µm

$(0.651)S_0$ = amount of solar radiation of wavelength < 0.9 µm

Calculate u^* according to Smith's (1966) method :-

$$u^* = \exp[0.07074\, T_d + \mathrm{T}] \qquad \textbf{eq. 3.7.4c}$$

where T_d = dew temperature

T = -0.02290 April - June

T = 0.02023 all other months

Calculate cloud cover adjustment factor according to Haurwitz's method (1948):-

$$CA = (a/94.4).\exp[-m(b-0.059)] \qquad \textbf{eq. 3.7.4d}$$

where CA = cloud adjustment factor

a&b = empirical coefficients dependent upon cloud type (Balick et al., 1981)

m = secant of the solar zenith angle

Calculate the energy reaching the surface according to Pochop et al., (1968):-

$$S_c = S_a - [S_a - (S_a . CA)] . CC^2 \qquad \textbf{eq. 3.7.4e}$$

where S_c = energy reaching the surface

S_a = energy reaching surface with no cloud

CA = cloud adjustment factor

CC = visual cloud cover in tenths

Calculate the effective incident net insolation according to:-

$$S = S_c . SF \qquad \textbf{eq. 3.7.4f}$$

where S = effective incident net insolation

S_c = net insolation

SF = slope factor

Calculate the slope factor, SF, according to :-

$$SF = \cos(z).\cos(SI) + \sin(z).\sin(SI).\cos(SAZ-SIAZ)$$

eq. 3.7.4g

where SF = slope factor

z = solar zenith angle

SI = slope of the surface

SAZ = solar azimuth angle

SIAZ = azimuth of the slope

Calculate the thermal infra-red energy inputs according to Sellers, (1965):-

$$I_{\surd 0} = \mathcal{E}\eth T_a^4 \left[c + b(e_a^{0.5}) \right]$$

eq. 3.7.4h

where $I_{\surd 0}$ = thermal infra-red energy input

$\mathcal{E}$ = emissivity (assumed to be 1)

$\eth$ = Stephan - Boltzman constant

T_a = shelter air temperature (kelvin)

e_a = water vapor pressure (mb)

b = empirical constant = 0.05

c = empirical constant = 0.61

Calculate the water vapor pressure, e_a ,according to Murray, (1967):-

$$e_a = RH. (6.108).exp(A.T_a)/(T_a+273.15-B)$$

eq. 3.7.41

where e_a = water vapor pressure

RH = relative humidity (decimal)

T_a = shelter air temperature (kelvin)

A = empirical constant = 17.269

B = empirical constant = 35.86

Calculate the infra-red radiation at surface as affected by cloud presence, according to Sellers, (1965) :-

$$I_{\surd t} = I_{\surd 0} (1+CIR . CC^2)$$

eq. 3.7.4j

where $I_{\surd t}$ = cloud adjusted infra-red radiation at surface

$I_{\surd 0}$ = infra-red radiation at surface with no cloud cover

CIR = coefficient dependent upon cloud type (Sellers, 1965 or Oke, 1978)

CC = cloud cover in tenths

Calculate the ground radiative emittance, $I\hat{}_t$, according to :-

$$I\hat{}_t = \mathcal{E}_g \delta (T_g)^4$$

eq. 3.7.4k

where $I^{\wedge}_{t}$ = ground radiative emittance from the surface

$\mathcal{E}_g$ = emissivity of the ground

δ = Stephan - Boltzman constant

T_g = ground temperature

Calculate the total infra-red input, I, to the surface according to:-

$$I = I_{\surd t} - I_{-t} \qquad \textbf{eq. 3.7.41}$$

where I = total thermal infra-red input to the surface

$I_{\surd t}$ = cloud adjusted infra-red radiation at the surface

I_{-t} = energy radiated from the surface

Calculate the conductive and convective sensible heat transfer, H, according to Lamb, (1974); Oke, (1978) :-

$$H = -\rho C_p k^2 z^2 \frac{\partial \Theta}{\partial z} \frac{\partial v}{\partial z} \text{ SCF} \qquad \textbf{eq. 3.7.4m}$$

where H = sensible heat transfer

ρ = air density

C_p = specific heat of dry air at constant pressure

k = von Karman's constant = 0.40

z = observation height

$\partial\theta / \partial z$ = partial derivative of potential temperature w.r.t. height

$\partial v / \partial z$ = partial derivative of windspeed w.r.t. height

SCF is defined by:-

eq. 3.7.4n

$$SCF = 1.175(1-15R_i)^{0.75} \quad \text{when } R_i \leq 0$$

$$SCF = (1-5R_i) \quad \text{when } 0 < R_i \leq 0.2$$

$$SCF = 0 \quad \text{when } R_i > 0.2$$

Calculate potential temperature, θ, according to:-

$$\theta = T_a \left(\frac{1000}{P} \right)^{0.286}$$

eq. 3.7.4o

where θ = potential temperature

T_a = air temperature

P = air pressure

Calculate the Richardson number, R, according to:-

$$R_i = \left(\frac{g}{\theta} \frac{\partial\theta}{\partial z} \right) / \left(\frac{\partial v}{\partial z} \right)^2$$

eq. 3.7.4p

where R = Richardson's number

g = gravity

Θ = average potential temperature between the surface and height, z

v = average wind velocity between the surface and height, z.

Calculate the latent heat exchange, Lx, according to:-

$$Lx = -\rho L k^2 z^2 \left(w \frac{\partial q}{\partial z} \right) \left(\frac{\partial v}{\partial z} \right) SCF \qquad \textbf{eq. 3.7.4q}$$

where Lx = latent heat exchange

ρ = air density

L = latent heat of evaporation

q = specific humidity

v = wind velocity

z = height (i.e. shelter height)

k = von Karman's constant = 0.40

w = saturation factor

SCF = defined in eq. **3.7.4n**

N.B.

Assume that heat flux from the ground, G, in temperate latitudes = 0

Calculate the following substitutions:-

SUBST. eq. **3.7.4c** into eq. **3.7.4b** to solve eq. **3.7.4b** = S_a

SUBST. eq. **3.7.4d** & eq. **3.7.4b** into eq. **3.7.4e** = S_c

SUBST. eq. **3.7.4e** & eq.**3.7.4g** into eq **3.7.4f** = S

SUBST. eq. **3.7.4h** into eq. **3.7.4j** = $I_{\sqrt{t}}$

SUBST. eq. **3.7.4j** & eq. **3.7.4k** into eq. **3.7.4l** = I

SUBST. eq. **3.7.4o** into eq. **3.7.4p** = R_i

SUBST. eq. **3.7.4p** into eq. **3.7.4n** = SCF

SUBST. eq. **3.7.4o** & eq. **3.7.4n** into eq. **3.7.4m** = H

SUBST. eq. **3.7.4o** into eq. **3.7.4q** = Lx

Substitute equations **3.7.4f**
3.7.4l
3.7.4m
3.7.4q into eq. **3.7.4a**
to solve for evaporation.

4 IMPROVED BSSS MODEL COMPONENTS

4.1 Soil Strength

4.1.1 Objectives

Off-road trafficability is a description of the ground surface state applied to movement over it. As discussed in **section 2**, application may range from agricultural, e.g. work days, animal grazing, etc., through civil, e.g. logging operations, to military, e.g. best route scenario, speed prediction, etc. For any of these applications, a standard quantitative method of expressing trafficability must be adopted.

The two major methods used to express soil strength are the 'Rating Cone Index', (RCI) and the 'Californian Bearing Ratio', (CBR). For the purpose of this project soil strength will be calculated in terms of RCI as existing off-road trafficability models, e.g. SMSP, predict values of RCI. **Figure 4.1** demonstrates the definition of RCI as a function of the shear soil strength. The RCI is the cone index (CI) that will result under traffic and is calculated according to :-

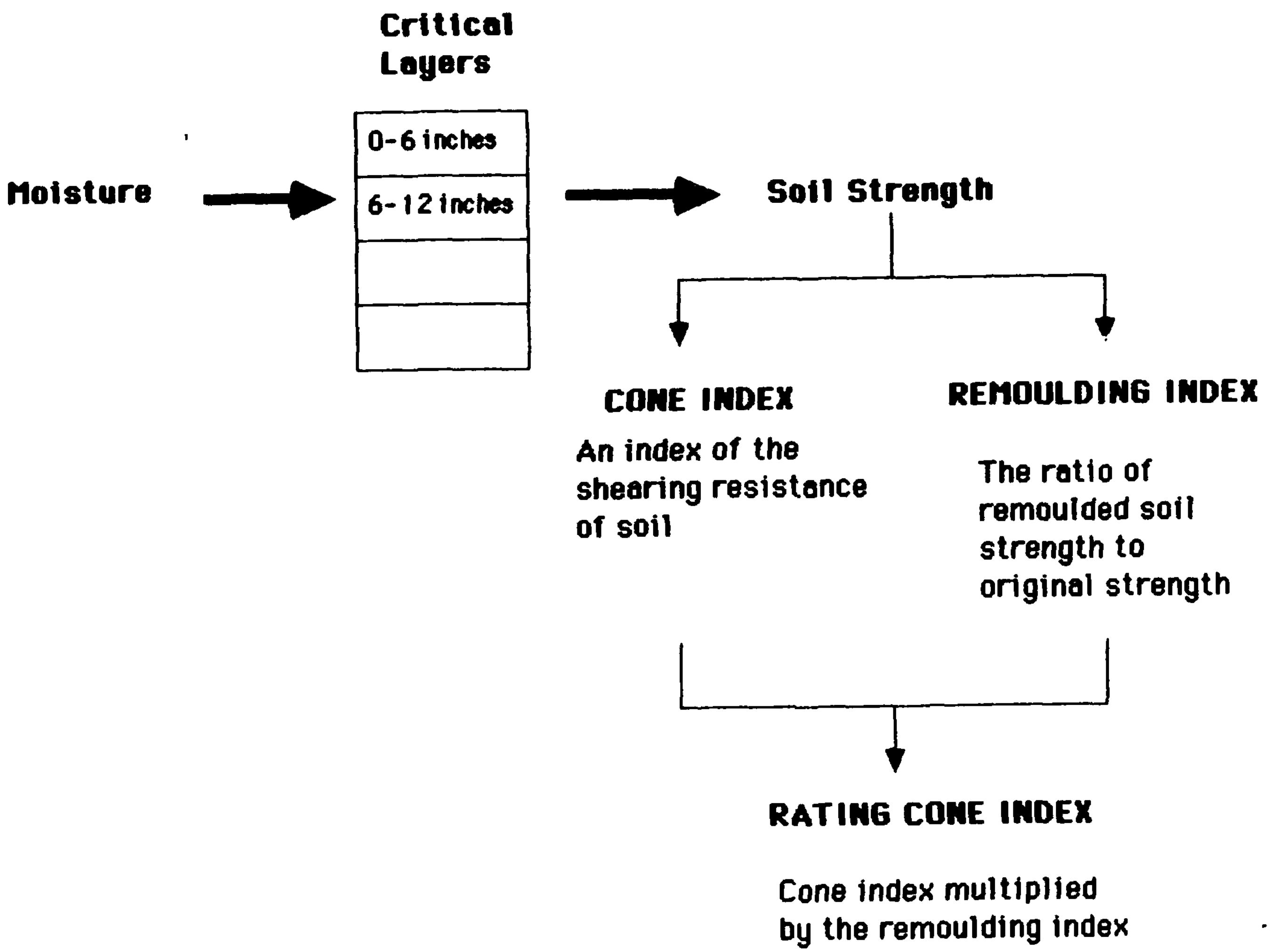
RCI = CI x Remoulded Index

e.g. CI = 85, Remoulded Index = 0.7

=> RCI = 85 x 0.7 = 59.5 (Department of the Army, 1959)

This project uses a remoulded index of 0.7 to calculate RCI from CI and the implications of this assumption are further examined in **section 8**.

FIGURE 4.1 DEFINITION AND RELATIONSHIPS OF SOIL STRENGTH TERMINOLOGY


Moisture
Critical Layers
0-6 inches
6-12 inches
Soil Strength
CONE INDEX
An index of the shearing resistance of soil
REMOULDING INDEX
The ratio of remoulded soil strength to original strength
RATING CONE INDEX
Cone index multiplied by the remoulding index

The shear soil strength is statically dependent of the mechanical characteristics of the soil, but is dynamically dependent on the soil moisture content. For application to trafficability the critical layers of the soil profile are 0-6 and 6-12 inches (Smith and Meyer, 1973).

As comparisons between the SMSP, VSAS2 and the BSSS will be required to assess the new schemes (**section 10.3**) it is necessary that they predict the same type of outputs. It is also hoped to apply some results to the Ministry of Defence's DRIVEB (**section 10.3**) model which requires an input of RCI.

4.1.2 Detailed Requirements

Having decided that the quantity we would like to predict is RCI, it is now possible to refine the general requirements stated in **section 1.2**:-

4.1.2a Soil strength is to be calculated in terms of RCI.

4.1.2b The scheme adopted or developed should preferably be physically based.

4.1.2c The scheme should be parsimonious with respect to data requirements.

4.1.2d The scheme should include the effect of soil moisture content on soil strength.

4.1.3 Review of Potential Soil Strength Schemes

In a review of soil trafficability prediction undertaken by the U.S. Corps of Engineers (1967) the soil moisture - soil strength empirical relationships of Collins (1967) and Moltham (1967) were highlighted. This work was the basis of an empirical prediction of RCI by Collins (1971). This was tne methodology used by both the SMSP scheme and Anderson (1983), and is therefore reviewed below along with the only physically based RCI calculation available.

4.1.3a Empirical RCI Calculation

Collins (1971) developed the following relationship to calculate RCI:-

$$\ln RCI = 4.605 + \frac{2.123 + 0.008(C) - 0.693 \ln M}{0.149 + 0.002(C)} \qquad \textbf{eq. 4.1.3a}$$

where M = moisture content (% dry weight)
C = percentage clay

As was discussed above, this relationship was the best there was for many years, and as such was used in the SMSP scheme and Anderson's prototype scheme. It is entirely empirical, and as such does not fulfil the requirements stated in **section 4.1.2**. It is therefore proposed to investigate any available physically based schemes.

4.1.3b Physically Based RCI Calculation

A model developed by Rohani and Baladi (1981) correlates cone index (CI) with the fundamental engineering properties of a soil. The scheme is based on modelling the penetration of a standard WES cone penetrometer (**figure 4.2**) into soil:-

$$CI = -C\cot\phi + \frac{2\tan\alpha(1+\sin\phi)G^m}{\{\gamma(D/2)\}^2\tan^3\phi}\;\frac{[3(\tan\alpha+\tan\phi)]}{3-\sin\phi}\;\Omega$$

eq. 4.1.3.1

where

$$\Omega = \frac{[C+\gamma(Z+L)\tan\phi]^{3-m} - [C+\gamma(Z+L)\tan\phi + (2-m)\gamma L\tan\phi](C+\gamma Z\tan\phi)^{2-m}}{(2-m)(3-m)}$$

eq. 4.1.3.2

and

$$m = 4\sin\phi / 3(1+\sin\phi)$$ **eq. 4.1.3.3**

CI = cone index

Z = depth from surface

C = soil cohesion

ϕ = angle of internal friction

γ = soil density

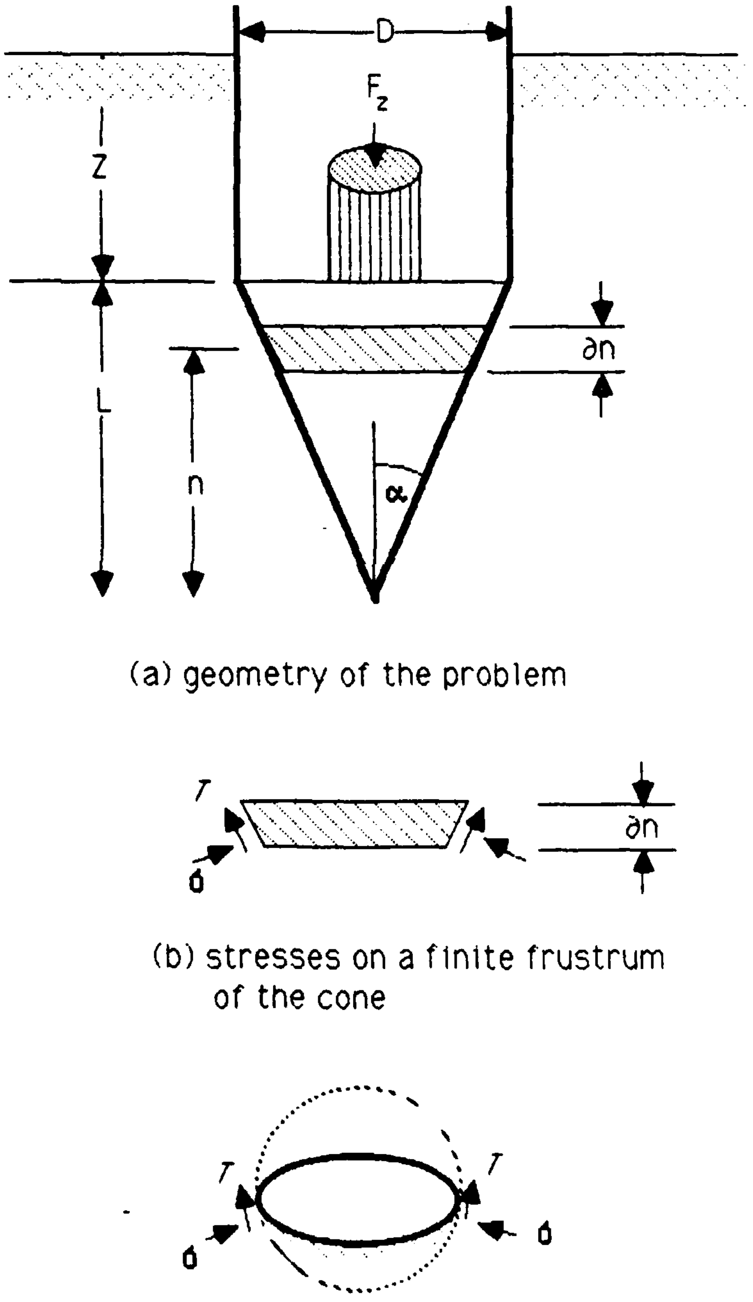
D = cone diameter

L = cone length

α = half the cone apex angle

G = apparent shear modulus

FIGURE 4.2 : THE CONE PENETRATION PROBLEM (FROM ROHANI AND BALADI, 1981)



(a) geometry of the problem

(b) stresses on a finite frustrum of the cone

(c) Analogy between cavity expansion and cone penetration process

Rohani and Baladi integrated the stresses of the cone penetrating the soil over the cone surface to produce this expression for CI. This derivation was carried out under fully drained conditions and therefore the relationship was unable to include the effect of soil water.

There were three ways inwhich variable pore pressures (both positive and negative) could be included in the Rohani and Baladi scheme and were discussed with Baladi (1987):-

Simple Alteration of 'Z' Term

In the original Rohani and Baladi relationship (**eq. 4.1.3.2**), Z is the 'depth' term with no implied treatment of pore pressure. Therefore, at the point of interest, ψ is assumed to be zero, i.e. at the water table. If hydrostatic conditions are assumed, as in **figure 4.3**, CI can be calculated at a point above or below a water table. The Z term in **equation 4.1.3.2** is replaced with $(Z + \psi)$ and the new situation is shown in **figure 4.4**.

The original Rohani and Baladi situation is shown in **figure 4.4a** and the proposed situation in **figure 4.4b**. The proposed situation shows a point of interest, @, where the soil suction, ψ, is known. Hydrostatic conditions (**figure 4.3**) are assumed and therefore the water table is ψ inches below point @. Therefore to calculate CI at a point @, Z in **equation 4.1.3.2** is replaced by $(Z + \psi)$.

This treatment is easy to include and was incorporated as a first attempt to include ψ in the calculation of CI The major problem with this treatment, is that it is no longer a derivable physical relationship and assumes hydrostatic conditions above and below the water table

FIGURE 4.3 DIAGRAM OF PIEZOMETRIC HEAD, φ, AND PRESSURE HEAD p/ρ IN THE HYDROSTATIC STATE

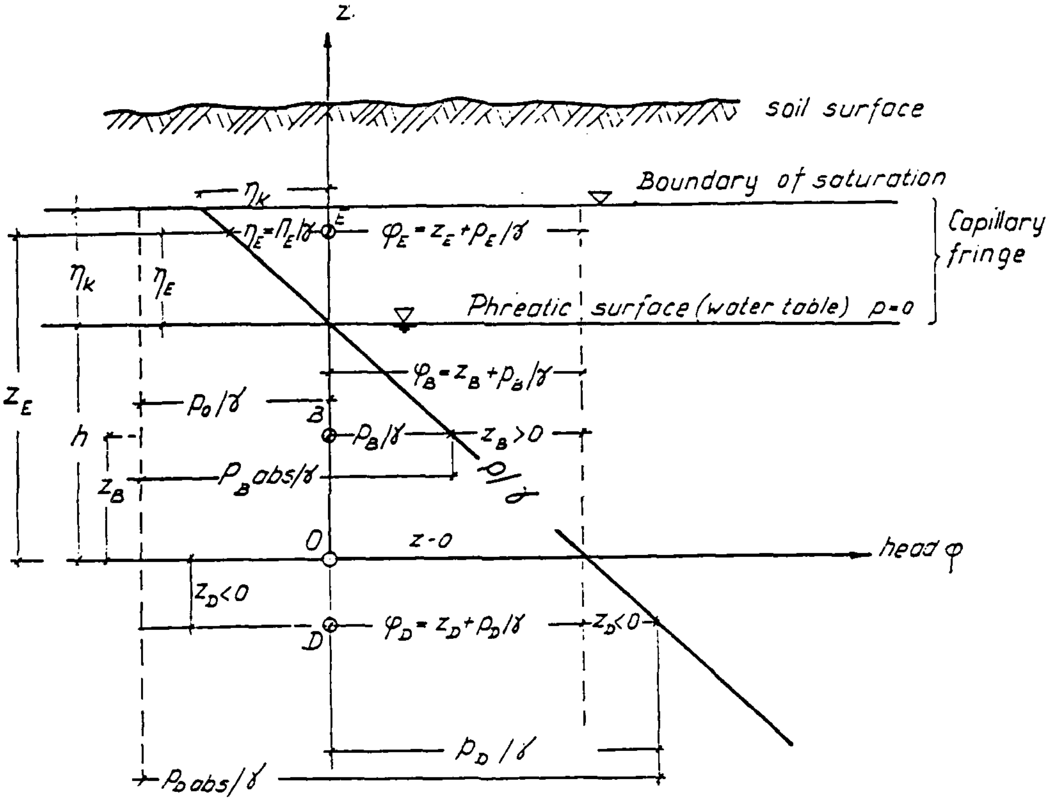
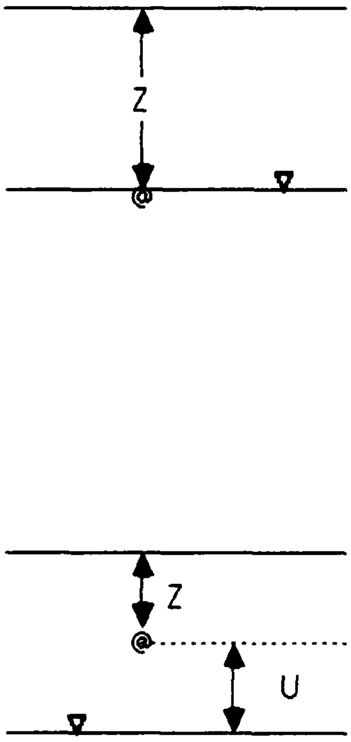


FIGURE 4.4: ALTERATION OF 'Z' TERM



(a) Original Rohani and Baladi Situation

(b) assumes hydrostatic conditions; water table is U inches below @, where U is thesoil suction at point @.

Rederivision of the Rohani and Baiadi Relationship Using The Effective Stress Equation

The fundamental equations used for deriving the Rohani and Baiadi relationship were as follows:-

By definition, CI is given as

$$CI = 4F_z/\pi D^2 \qquad \textbf{eq. 4.1.3.4}$$

where the resistive force, F_z, is given by

$$F_z = \int_0^L (\sigma \tan \alpha + \tau) 2\pi r \partial n \qquad \textbf{eq. 4.1.3.5}$$

where the shear stress, τ, is given by

$$\tau = C + \sigma \tan \phi \qquad \textbf{eq. 4.1.3.6}$$

and

r = radius of a finite frustum of the cone = $n \tan \alpha$
σ = normal stress
∂n = width of a finite frustum of the cone (**figure** 4.2)

Columb's relationship (**equation 4.1.3.6**) could be replaced by the effective stress relationship:-

$$s' = C' + (\sigma - \psi) \tan \phi' \qquad \textbf{eq. 4.1.3.7}$$

This treatment has the problem of relating s' and CI but indicates a possibility for further research.

Rederivation of the Rohani and Baladi Relationship Using The Fredlund Equation

Fredlund's (1978) approach is a theoretically sound estimation of soil shear strength under pore pressure conditions, which may be either positive or negative (**figures 4.4** and **4.5**).

Fredlund's equation is given by:-

$$\tau(n) = C + [\delta(n) - \psi_a] \tan \phi + (\psi_a - \psi_w) \tan \phi^b \qquad \textbf{eq. 4.1.3.8}$$

where

ψ_a = pore air pressure

ψ_w = pore water pressure

ϕ^b = friction angle; for given $(\delta_n - \psi_a)$ gradient between $(\psi_a - \psi_w)$ and strength (**figure 4.5**)

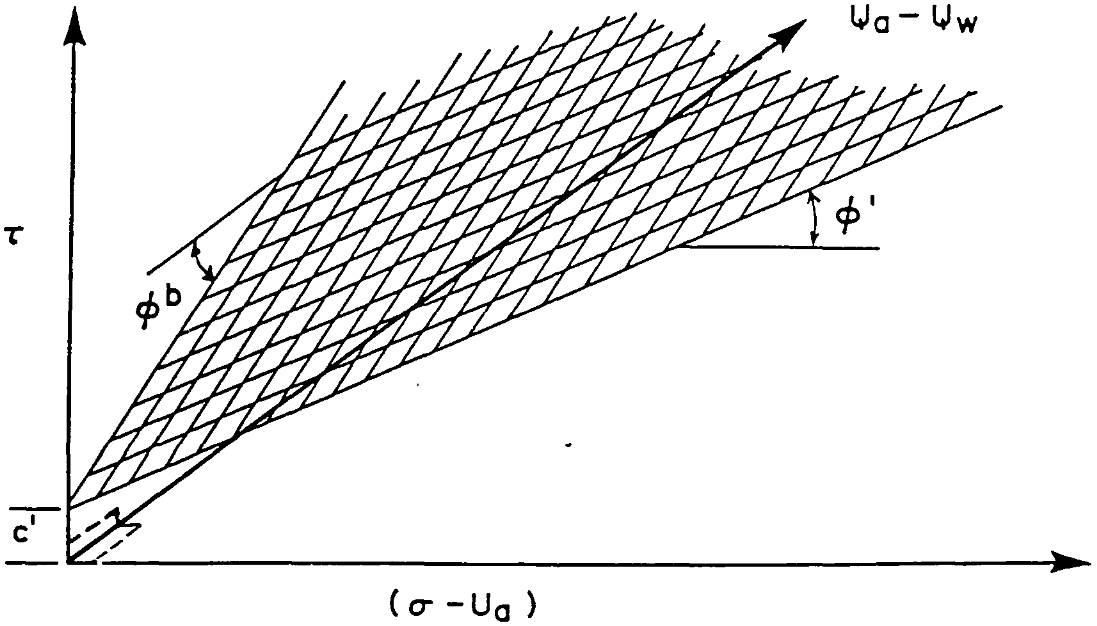
It would therefore be most appropriate to replace the Columb equation (**equation 4.1.3.6**) with the Fredlund equation and solve for CI. A full technical description of this method is given in **section 4.1.4** as the method to be used in the BSSS for calculating CI.

4.1.4 Technical Description of Rederived Rohani and Baladi Relationship Using Fredlund's Equation for Soil Shear Strength

The key equations from Rohani and Baladi (1981) are.

$$CI = 4F_z / \pi D^2 \qquad \textbf{eq. 4.1.4.1}$$

FIGURE 4.5 PLANAR SURFACE REPRESENTING THE SHEAR STRENGTH EQUATION FOR AN UNSATURATED SOIL



and

$$F_z = \int_0^L [\sigma(n)\tan\alpha + \tau(n)]\, 2\pi r(n)\partial n \qquad \text{eq. 4.1.4.2}$$

where the expression for the internal pressure for an expanding spherical cavity in an unbounded elastic-plastic medium is given by Vesic (1972):-

$$\sigma = 3(q + C\cot\phi)\frac{(1 + \sin\phi)}{(3 - \sin\phi)}\left(\frac{G}{(C + q\tan\phi)}\right)^m - C\cot\phi \qquad \text{eq. 4.1.4.3}$$

which can be used to define $\sigma(n)$ in **equation 4.1.4.2** as:-

$$\sigma(n) = 3(q(n) + C\cot\phi)\frac{(1 + \sin\phi)}{(3 - \sin\phi)}\left(\frac{G}{(C + q(n)\tan\phi)}\right)^m - C\cot\phi$$

eq. 4.1.4.4

and

$$r(n) = n\tan\alpha \qquad \text{eq. 4.1.4.5}$$

$$q(n) = (Z + L - n)\gamma \qquad \text{eq. 4.1.4.6}$$

Fredlund's Equation :-

$$\tau(n) = C + (\sigma(n) - \psi_a)\tan\phi + (\psi_a - \psi_w)\tan\phi^b \qquad \text{eq. 4.1.4.7}$$

$$m = 4\sin\phi \,/\, 3(1 + \sin\phi) \qquad \text{eq. 4.1.4.8}$$

For the calculation the following are constant:-

α, C, ϕ, G, $\bar{Z}$, L, γ

Solve $\sigma(n)\tan\alpha + \tau(n)$ through **equations 4.1.4.4** and **4.1.4.7**:-

$$\sigma(n)\tan\alpha + \tau(n) = C - \psi_a\tan\phi + (\psi_a - \psi_w)\tan\phi^b + \sigma(n)(\tan\alpha + \tan\phi)$$

eq. 4.1.4.9

Substitute **equation 4.1.4.9** into **equation 4.1.4.1**:-

$$F_z = 2\pi \int_0^L n\tan\alpha\,[C - \psi_a \tan\phi + (\psi_a - \psi_w)\tan\phi^b + \sigma(n)(\tan\alpha + \tan\phi)]\,\partial n$$

eq. 4.1.4.10

=>

$$= 2\pi\tan\alpha\,(C - \psi_a\tan\phi + (\psi_a - \psi_w)\tan\phi^b \int_0^L n\,\partial n$$

$$+ 2\pi\tan\alpha\,(\tan\alpha + \tan\phi) \int_0^L n\sigma(n)\partial n$$

=>

$$= \pi(C - \psi_a\tan\phi + (\psi_a - \psi_w)\tan\phi^b)L^2\tan\alpha$$

$$+ 2\pi\tan\alpha\,(\tan\alpha + \tan\phi) \int_0^L n\sigma(n)\partial n$$

=>

$$= \pi(C - \psi_a\tan\phi + (\psi_a - \psi_w)\tan\phi^b)L^2\tan\alpha$$

$$- \pi CL\tan\alpha\,(\tan\alpha\cot\phi + 1)$$

$$+ 2\pi\tan\alpha\,(\tan\alpha + \tan\phi) \int_0^L n\sigma'(n)\partial n$$

eq. 4.1.4.11

where $\sigma'(n) = \sigma(n) + C\cot\phi$ **eq. 4.1.4.12**

Expand **equation 4.1.4.4** by substituting **equation 4.1.4.12**:-

=>

$$\sigma'(n) = 3\left(q(n) + C\cot\phi\right)\left(\frac{1 + \sin\phi}{3 - \sin\phi}\right)\left(\frac{G}{C + q(n)\tan\phi}\right)^m \qquad \textbf{eq. 4.1.4.13}$$

Re-arrange **equation 4.1.4.8**

=>

$$\sin\phi = 3m / (4 - 3m) \qquad \textbf{eq. 4.1.4.14}$$

Re-arrange **equation 4.1.4.13**

=>

$$\sigma'(n) = \frac{3(1 + \sin\phi)\,G^m}{(3 - \sin\phi)} \qquad \frac{q(n) + C\cot\phi}{(q(n)\tan\phi + C)^m} .$$

=>

$$\frac{3(1 + \sin\phi)\,G^m}{(3 - \sin\phi)} \qquad \frac{q(n) + C\cot\phi}{[\tan\phi(q(n)\tan\phi + C\cot\phi)]^m}$$

=>

$$\frac{3(1 + \sin\phi)}{(3 - \sin\phi)}\ \frac{G^m}{(\tan\phi)^m} \qquad [q(n) + C\cot\phi\,]^{1-m}$$

Substitute **equation 4.1.4.6**

=>

$$\Sigma\,[\,\gamma(Z + L) + C\cot\phi - \gamma n]^{1-m} \qquad \textbf{eq. 4.1.4.15}$$

where

$$\Sigma = \frac{3(1 + \sin \phi)}{(3 - \sin \phi)} \; \frac{(\;G\;)^m}{(\tan \phi)^m}$$

hence

$$\int_0^L n\sigma'(n)\partial n = \Sigma \int_0^L n[\gamma(Z+L) + C \cot \phi - \gamma n]^{1-m} \partial n$$

=>

$$\Sigma\gamma^{1-m}\int_0^L n[Z+L+(C/\gamma)\cot \phi - n]^{1-m}\partial n$$

=>

$$\Sigma\gamma^{1-m}\int_0^L - [Z+L+(C/\gamma)\cot \phi - n - (Z+L+(C/\gamma)\cot \phi)]$$
$$\times [Z+L+(C/\gamma)\cot \phi - n]^{1-m}\partial n$$

=>

$$\Sigma\gamma^{1-m}(Z+L+(C/\gamma)\cot \phi)\int_0^L [Z+L+(C/\gamma)\cot \phi - n]^{1-m}\partial n$$
$$- \Sigma\gamma^{1-m}\int_0^L [Z+L+(C/\gamma)\cot \phi - n]^{2-m}\partial n$$

=>

$$\Sigma\gamma^{1-m}(Z+L+(C/\gamma)\cot \phi)\left[\frac{(Z+L+(C/\gamma)\cot \phi - n)^{2-m}}{-(2-m)}\right]_0^L$$
$$- \Sigma\gamma^{1-m}\left[\frac{(Z+L+(C/\gamma)\cot \phi - n)^{3-m}}{-(3-m)}\right]_0^L$$

=>

$$- \Sigma \gamma^{1-m} (Z+L+(C/\gamma)\cot \phi) \frac{(Z+(C/\gamma)\cot \phi)^{2-m}}{2-m}$$

$$+ \Sigma \gamma^{1-m} (Z+L+(C/\gamma)\cot \phi) \frac{(Z+L+(C/\gamma)\cot \phi)^{2-m}}{2-m}$$

$$+ \Sigma \gamma^{1-m} \frac{(Z+(C/\gamma)\cot \phi)^{2-m}}{3-m}$$

$$- \Sigma \gamma^{1-m} \frac{(Z+L+(C/\gamma)\cot \phi)^{3-m}}{3-m}$$

=>

$$- \frac{\Sigma \gamma^{1-m}}{2-m} (Z+L+(C/\gamma)\cot \phi) (Z+(C/\gamma)\cot \phi)^{2-m}$$

$$+ \Sigma \gamma^{1-m} (Z+L+(C/\gamma)\cot \phi)^{3-m} [1/(2-m) - 1/(3-m)]$$

$$+ \frac{\Sigma \gamma^{1-m}}{3-m} (Z+(C/\gamma)\cot \phi)^{3-m}$$

therefore =>

$$\int_0^L n\sigma'(n)\partial n = \frac{\Sigma \gamma^{1-m}}{3-m} (Z+(C/\gamma)\cot \phi)^{3-m}$$

$$+ \frac{\Sigma \gamma^{1-m}}{(2-m)(3-m)} (Z+L+(C/\gamma)\cot \phi)^{3-m}$$

$$- \frac{\Sigma \gamma^{1-m}}{2-m} (Z+L+(C/\gamma)\cot \phi) (Z+(C/\gamma)\cot \phi)^{2-m}$$

=>

$$\frac{\Sigma \gamma^{1-m}}{(2-m)(3-m)} (Z+L+(C/\gamma)\cot \phi)^{3-m}$$

$$- \frac{\Sigma \gamma^{1-m}}{(2-m)(3-m)} (Z+(C/\gamma)\cot \phi)^{3-m}$$

$$- \frac{\Sigma \gamma^{1-m}}{2-m} L(Z+(C/\gamma)\cot \phi)^{2-m}$$

eq. 4.1.4.16

Now note that since $m = \frac{4\sin \phi}{3(1+\sin \phi)}$

and hence $\sin \phi = \frac{3m}{4-3m}$

=>

$$\frac{3(1+\sin \phi)}{3-\sin \phi} = \frac{4\sin \phi}{m(3-\sin \phi)} = \frac{4 \times 3m}{4-3m} \times \frac{1}{m} \times \frac{1}{(3-3m)/(4-3m)}$$

$$= \frac{12}{4-3m} \times \frac{4-3m}{12-9m-3m} = \frac{1}{1-m}$$

therefore

$$\Sigma = \frac{1}{1-m} (G \cot \phi)^{m}$$

eq. 4.1.4.17

Substitute **equations 4.1.4.16** and **4.1.4.17** into **equation 4.1.4.11**

$$F_z = \frac{2\pi \tan\alpha(\tan\alpha + \tan\phi)}{(1-m)(2-m)(3-m)}(G\cot\phi)^m \gamma^{1-m}$$

$$\times \left\{(Z+L+(C/\gamma)\cot\phi)^{3-m} - (Z+(C/\gamma)\cot\phi)^{3-m} - (3-m)L(Z+(C/\gamma)\cot\phi)^{2-m}\right\}$$

$$-\pi(C\tan\alpha\cot\phi - \psi_a\tan\phi + (\psi_a - \psi_w)\tan\phi^b)L^2\tan\alpha$$

eq. 4.1.4.18

Substitute **equation 4.1.4.18** into **equation 4.1.4.1** and therefore the rederived Rohani and Baladi relationship using the Fredlund equation may be expressed as:-

$$CI = \frac{4\times 2\pi\tan\alpha(\tan\alpha + \tan\phi)}{\pi D^2(1-m)(2-m)(3-m)}(G\cot\phi)\, g^{1-m}$$

$$\times \left\{(Z+L+(C/\gamma)\cot\phi)^{3-m} - (Z+(C/\gamma)\cot\phi)^{3-m} - (3-m)L(Z+(C/\gamma)\cot\phi)^{2-m}\right\}$$

$$-\frac{4\pi}{\pi D^2}(C\tan\alpha\cot\phi - \psi_a\tan\phi + (\psi_a - \psi_w)\tan\phi^b)L^2\tan\alpha$$

=>

$$\frac{8\tan\alpha(\tan\alpha + \tan\phi)}{D^2(1-m)(2-m)(3-m)}(G\cot\phi)\gamma^{1-m}$$

$$\times \left\{(Z+L+(C/\gamma)\cot\phi)^{3-m} - (Z+(C/\gamma)\cot\phi)^{3-m} - (3-m)L(Z+(C/\gamma)\cot\phi)^{2-m}\right\}$$

$$-\frac{4}{D^2}(C\tan\alpha\cot\phi - \psi_a\tan\phi + (\psi_a - \psi_w)\tan\phi^b)L^2\tan\alpha$$

eq. 4.1.4.19

4.2 Albedo

4.2.1 Introduction

The reflection properties of natural surfaces are of interest in meteorologically related problems because of their influence on the radiation budget of the atmosphere. Measurements of reflection on the field scale have usually been made by pyranometers, producing a range of albedo (i.e. the ratio of outgoing to incoming radiation of a surface) for a 'typical' vegetated surface. Models to predict the albedo of a surface from its physical characteristics, such as the Seller's two stream approximation model (1972), have proved both complex and difficult to validate.

This section investigates why an estimation of albedo is important in the BSSS, the accuracy to which albedo can be estimated from existing data for various surfaces, and, the effect of the maximum possible error induced by the albedo input parameter.

4.2.2 Background

The reflection of energy from vegetated surfaces is strongly anisotropic in behaviour. Providing the vegetated surface is snow and ice free, the variation of reflection is dependent on six primary variables (Oke, 1979; Kriebel, 1977): locality, time, vegetation characteristics, leaf wetness, wavelength, amount and angular distribution of the incident radiation.

The calculation of evaporation by a non-isothermal method (Khale, 1977) has been described in **section 3.4**, where the key relationship was described by:-

$$S = (1-\alpha_g)\,[1-A(u^*,z)](0.349)S_0 \cos z \qquad \textbf{eq. 20211g.2}$$

$$+ (1-\alpha_g)\,[(1-\alpha_0)/(1-\alpha_0\underline{\alpha}_g)](0.651)S_0 \cos z$$

where S = incoming radiation at the ground with no cloud cover

α_g = surface albedo

$\underline{\alpha}_g$ = average ground albedo

α_0 = atmospheric albedo for Raleigh scattering , equal to

$0.085 - 0.247 \log_{10} [(p_s/p_o)\cos z]$

p_s = surface pressure

p_o = 1000 mb

z = zenith angle of the sun as a function of the time of day and year

$A(u^*,z)$ = Mugge - Moller absorption function, equal to $0.271(u^* \sec z)^{0.803}$

u^* = effective water vapour content of the atmosphere

S_0 = solar radiation incident on top of the atmosphere

$(0.349)S_0$ = amount of solar radiation of wavelength > 0.9 µm

$(0.651)S_0$ = amount of solar radiation of wavelength < 0.9 µm

From **equation 2021 Ig.2**, three albedo terms are required - α_g, $\underline{\alpha}_g$ and α_0. The determination of α_0 is well established (Khale, 1977) but the values of α_g and $\underline{\alpha}_g$ are more problematic and are discussed below.

The influence of a variation in α_g on total evaporation is shown in **figure 4.6**. This degree of sensitivity implies that when evaporation is the predominant process affecting the top soil layers, the value of albedo may be important in RCI calculations. A sensitivity analysis (**figure 4.7**) of RCI to albedo indicates that providing the total range of albedo for each type of surface is +/- 12%, the variation in RCI will not exceed 10%. This will enable the empirical tables of albedo for different vegetations (**table 4.1**) to be used with some confidence. The possible variations of albedo in excess of the tabulated ranges is discussed in **section 4.2.3**.

4.2.3 Major Contributors to Albedo Variation

4.2.3.1 Locality and Time

Site location may be described in terms of relative solar altitude. **Figure 4.8** shows results for grass, kale, oak(bare and leaved), spruce and pine taken over various solar altitudes. The variation of solar altitude throughout the day and throughout the year may have a serious influence on albedo and hence on RCI calculations. The following table indicates the magnitude of variation expected:

FIGURE 4.6:EFFECT OF EVAPORATION ON CONE INDEX CALCULATION

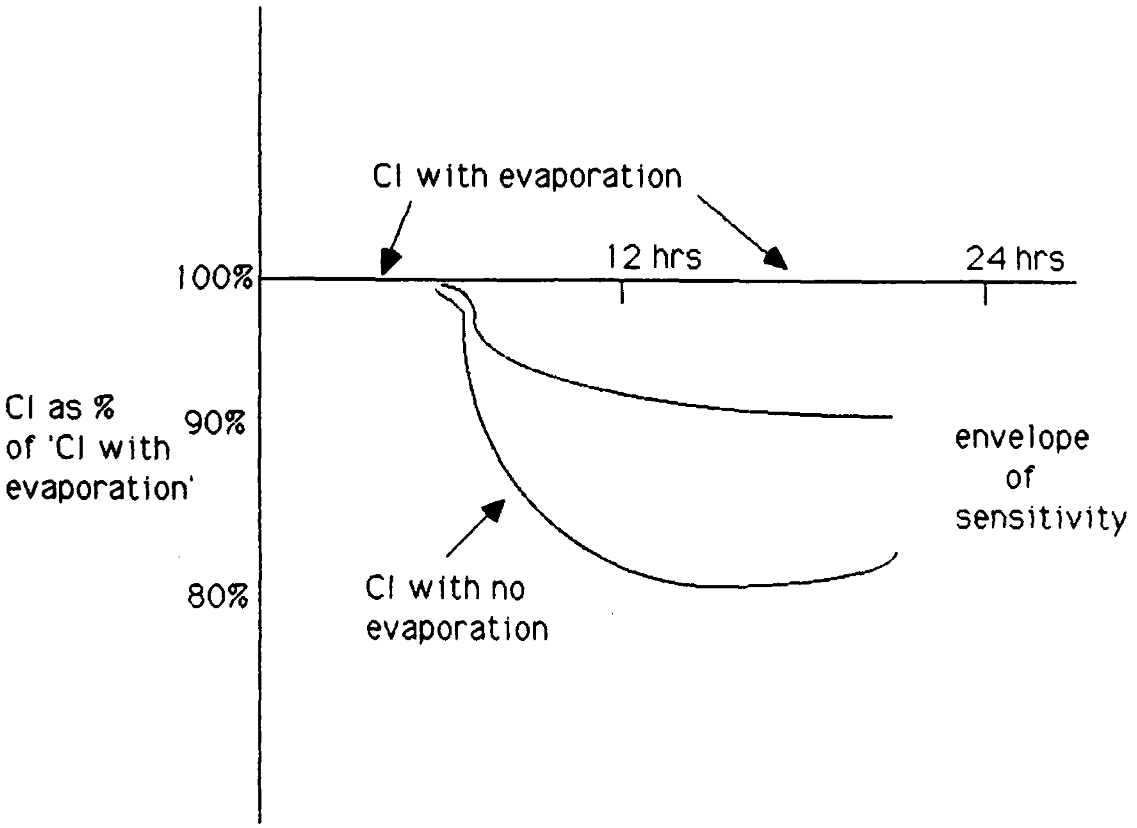


FIGURE 4.7: RELATIONSHIP BETWEEN ALBEDO AND RCI FOR DIFFERENT SOILS

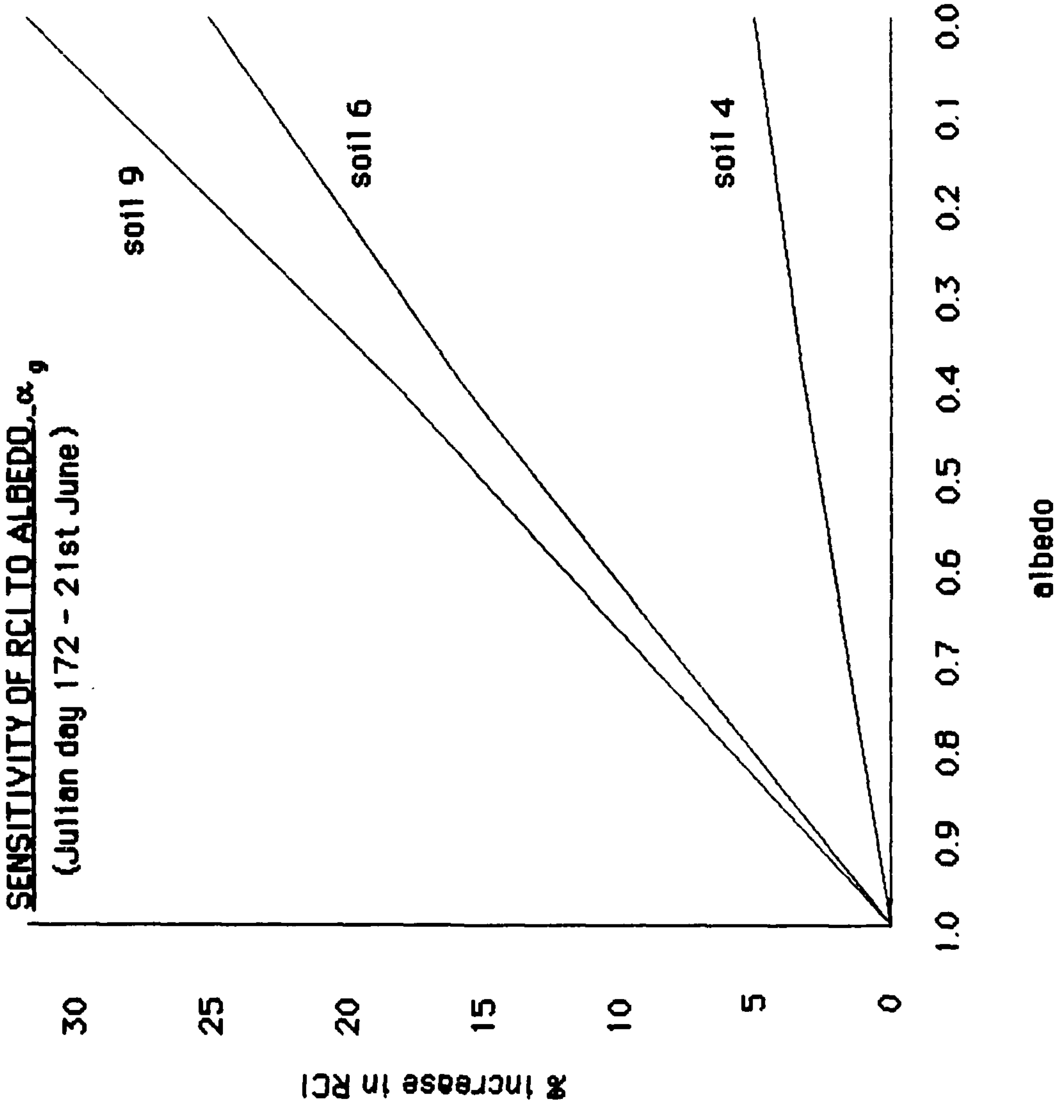
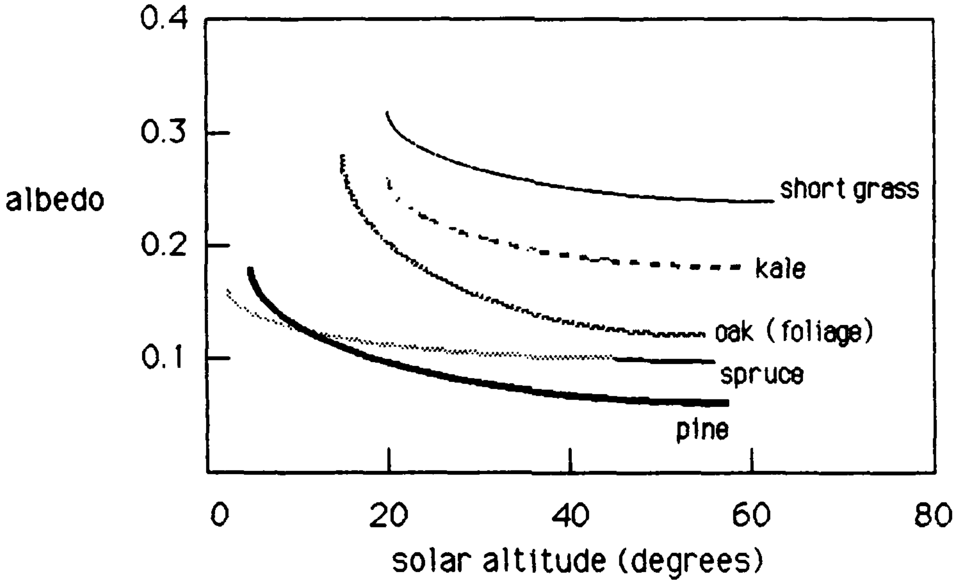


TABLE 4.1 Radiative Properties of Natural Surfaces
(From Sellers,1965; list, 1966; Paterson, 1969; and
Monteith, 1973)

SURFACE	REMARKS	ALBEDO	EMISSIVITY
		α	ε
Soils	Dark, wet Light, dry	0.05-0.40	0.90-0.98
Desert		0.20-0.45	0.84-0.91
Grass	Long (1.0m) Short (0.02m)	0.16- 0.26	0.90- 0.95
Agricultural Crops Tundra		0.18-0.25	0.90-0.99
Orchards		0.15-0.20	
Forests			
Deciduous	Bare Leaved	0.15- 0.20	0.97- 0.98
Coniferous		0.05-0.15	0.97-0.99
Water	Small zenith angle Large zenith angle	0.03-0.10 0.10-1.00	0.92-0.97 0.92-0.97
Snow	Old Fresh	0.40- 0.95	0.82- 0.99
Ice -	Sea Glacier	0.30-0.45 0.20-0.40	0.92-0.97

FIGURE 4.8: EFFECT OF SOLAR ALTITUDE ON ALBEDO



Relation between the albedo of vegetation and solar altitude on sunny days. Grass & kale (Monteith and Szeicz,1961); oak forest (Rauner, 1976); spruce forest (Jarvis et al., 1976); scots pine forest (Stewart,1971)

SURFACE	RANGE	% CHANGE IN ALBEDO	PREDICTED % CHANGE IN RCI
Short grass	.30-.24	6%	2.4%
Kale	.27-.19	8%	3.2%
Oak (foliage)	.30-.16	14%	5.6%
Oak (bare)	.25-.13	12%	4.8%
Spruce	.18-.12	6%	2.4%
Pine	.20-.08	12%	4.8%

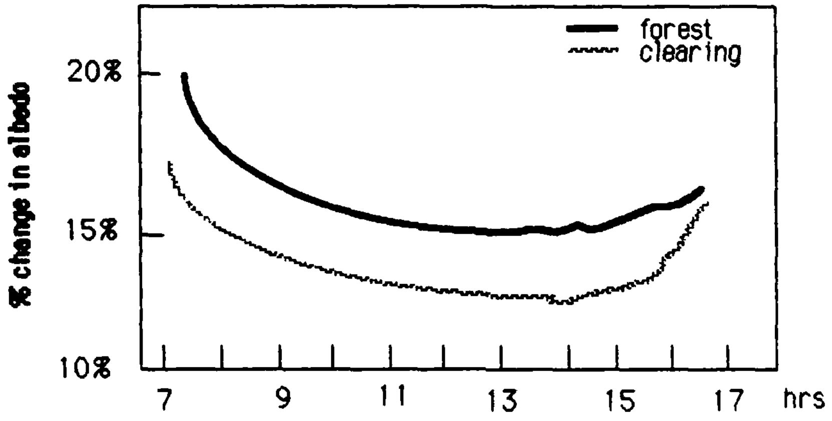
Figure 4.9 indicates that seasonal variations of solar altitude on albedo maybe small but that the daily variation may be important. The extent of the variation throughout the day is summarized in the following table:

SURFACE	RANGE	% CHANGE IN ALBEDO	PREDICTED % CHANGE IN RCI
Clearing	.195-.135	6%	2.4%
Forest	.166-.11	5.5%	2.2%

It is noticeable from **figure 4.9** that natural surfaces reflect strongly at low angles of incidence, i.e. the highest albedos occur at times of low energy input and it may therefore be said that the effect of this diurnal variation is small.

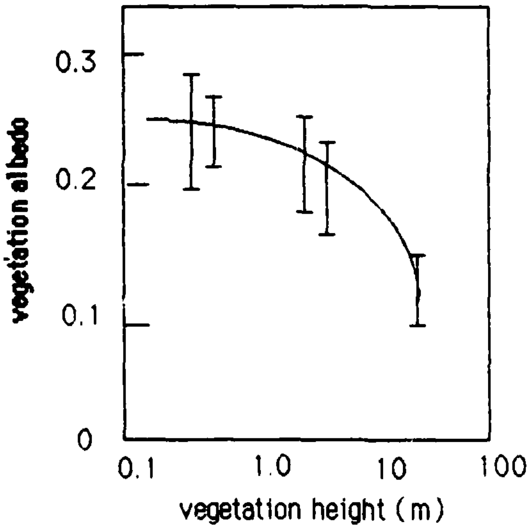
The results above indicate that the influence of locality and time on albedo and therefore on RCI calculations is relatively small. It will be the effect of combining all the separate small variations into a 'total possible variation' that shall be of interest and will be discussed below.

FIGURE 4.9: DIURNAL VARIATION OF ALBEDO



The diurnal variation of the average albedo during February, March, May, June, August and September.

FIGURE 4.10: EFFECT OF VEGETATION HEIGHT ON ALBEDO



Relation between the albedo of vegetation and its height. Vertical lines are two standard deviations.
(After Stanhill, 1970)

4.2.3.2 vegetation Characteristics

The influence of vegetation characteristics such as leaf orientation and height have been shown by Oke (1978) to have an effect on the albedo of a vegetated surface. The scale on which we are wanting to consider albedo is at the field scale and therefore the effect of leaf orientation shall be considered negligible. The effect of vegetation height at this scale on albedo can be seen in **figure 4.10** . From this information the grass, small bush and tree canopies have been assessed as follows:

SURFACE	RANGE	% CHANGE IN ALBEDO	PREDICTED % CHANGE IN RCI
Grass	0	0%	0%
Bushes (0.5-2.0 m)	.255-.23	2.5%	0.8%
Trees (1-8 m)	.25-.17	8%	3.2%

From these calculations it can be seen that when considering trees, the height of them could be important in albedo estimation. As trees are a difficult type of vegetation to consider because of the interception and transpiration which are not allowed for as yet in the BSSS, it is therefore proposed that the small effect that variations in vegetation height might induce on other vegetated surfaces may be considered almost negligible

4.2.3.3 Leaf Wetness

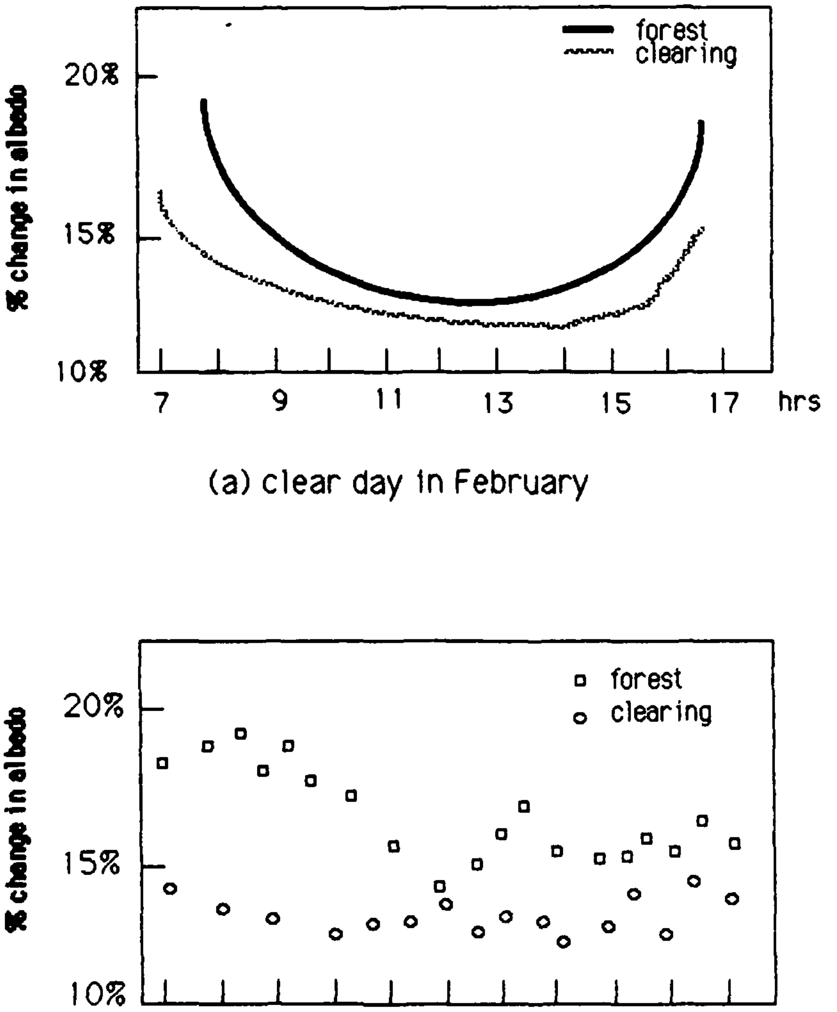
Oke (1978) and Monteith (1961) found leaf wetness to be significantly influential in their albedo measurements. Greater reflection from short grass occurred when wet - either after rain or a dewy night. Lockwood (1985) estimates that a difference of 6-8% in albedo measurements was observed between moist and dry soil. This difference would be seen as a 2.4-3.2% change in the RCI calculation.

When leaf wetness occurred after a dewy night, it could be expected that the reflection might be similar to that of a water surface. **Table 4.1** shows some 'typical' albedo values for a water surface which indicates that the influence of wet leaves could be very important at low zenith angles, but as this is when the solar input is at its weakest the degree of influence is significantly reduced.

4.2.3.4 Amount and Angular Distribution of Incident Radiation

The diurnal variation of radiation can be seen in **figure 4.11**. The hours between 0600 and 1800 can be considered as the most important in evaporation calculations. This is also therefore the time period to consider the reflection properties of different surfaces. The category of radiation is also important - either direct of diffuse, and as can be seen in **figure 4.11** the difference in albedo and the effect on RCI calculations can be summarized as follows:

FIGURE 4.11 DIURNAL VARIATION OF ALBEDO



(a) clear day in February

(b) cloudy day in September

SURFACE	RANGE	% CHANGE IN ALBEDO	PREDICTED % CHANGE IN RCI
Forest (clear)	.187-.126	6.1%	2.44%
Forest (cloudy)	18-.14	4%	1.6%
Clearing (clear)	.21-.116	9.4%	3.8%
Clearing (cloudy)	.142-.128	1.4%	0.6%

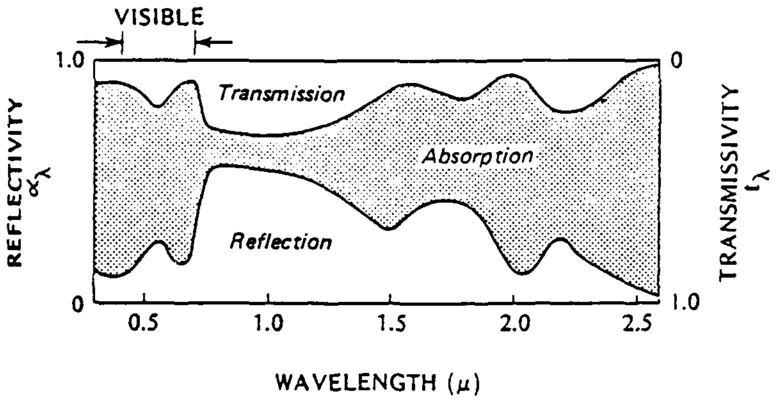
While the range of albedo measurements on a cloudy day is reduced from that on a clear day, the average albedo for the day is not significantly altered. As the radiation during an overcast dav is richer in diffuse radiation, this result supports the theory that the diurnal variation of the albedo is primarily a result of the nature of the reflection of the direct component of the solar radiation.

4.2.3.5 Wavelength

The energy emitted by the sun and impinging upon the Earth's surface has a large range of wavelengths, i.e. 0.3 μm - several metres. The region of this spectrum which is of interest to this application is 0.5-2.2 μm as this is the energy 'available' for evaporation (**figure 4.12**).

There are many tables such as **table 4.1** , which detail results of albedo measurements as a range within the 'typical' albedo of a surface may be expected to fall. These values are integrated over the 0.5-2.2 μm waveband and are therefore suitable for the estimations required in the BSSS.

FIGURE 4.12 RELATIONSHIP BETWEEN WAVELENGTH AND ALBEDO



. Idealized relation between wavelength and the reflectivity (α), ransmissivity (t) and absorptivity (a) of a green leaf (after Monteith, 1965).

From **equation 2021 Ig.2** it can be seen that the equation is split into two parts according to wavelength, i.e. it is defined that 65.1 % of the relevant solar radiation has a wavelength less than 0.9 μm, and therefore that 34.9 % of relevant solar radiation has wavelengths greater than 0.9 μm

Work carried out by Kriebel (1977) determined that variation of reflection with wavelength is often marked (**figure 4.12**). There are several methods which attempt to define reflection quantities, such as, the *spectral bidirectional reflectance distribution function, Nicodemus* (1970), and Kasten and Rascke (1974); or the spectral biconical reflectance factor, Kriebel (1977). The methodology of Nicodemus involves conical geometry and eighteen possible reflection properties, whereas Kriebel's spectral biconical reflectance factor was determined from measurements of albedo made over seven narrow spectral intervals (**table 4.2**). The methodology and detailed results are given in Kriebel (1977). **Table 4.3** provides an example of the data collected from four vegetation covers. The results have been divided into two parts as defined by **equation 2021 Ig.2** and the average albedos over these two parts of the spectrum have been calculated for use in a sensitivity analysis (**section 4.2.3.6**).

By considering albedo as a function of the wavelength, Kriebel (1978) was able to demonstrate (**figures 4.13 - 4.16**) that the variation of albedo with solar altitude at wavelengths greater than 0.9 μm was very small.

TABLE 4.2 Example of Biconical Reflectance Factors From Kriebel (1977)

Each table of the spectral reflectance factor is characterized by the surface type and wavelength (μm). The measurements are taken at various zenith angles (THETAR) and zenith angle of incidence (THETAI). To each pair of THETAR and THEATI belongs a block of seven values corresponding to the seven azimuths (PHI).

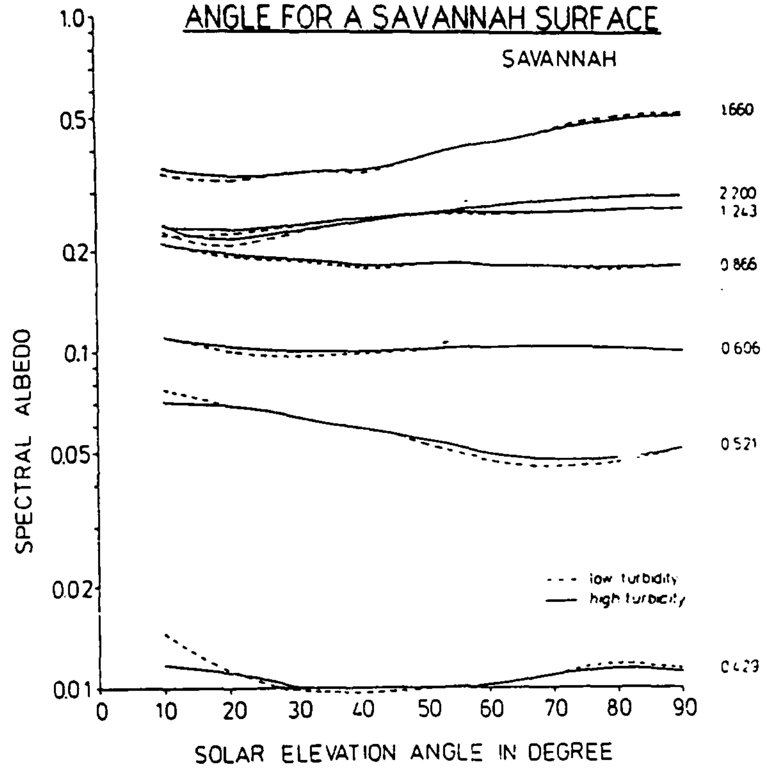
SURFACE TYPE: PASTURE

wavelength = 0.429 μm

THETAR		0	10	20	30	40	50	60	70	80	90
THETAI	PHI										
	0	.0197	.0196	.0195	.0183	.0189	.0193	.0188	.0186	.0139	.0141
	30	.0197	.0196	.0195	.0183	.0189	.0192	.0188	.0186	.0139	.0141
	60	.0197	.0196	.0195	.0183	.0189	.0192	.0188	.0186	.0139	.0141
0	90	.0197	.0196	.0195	.0183	.0189	.0192	.0188	.0186	.0139	.0141
	120	.0197	.0196	.0195	.0183	.0189	.0192	.0188	.0186	.0139	.0141
	150	.0197	.0196	.0195	.0183	.0189	.0192	.0188	.0186	.0139	.0141
	180	.0197	.0196	.0195	.0183	.0189	.0192	.0188	.0186	.0138	.0141
	0	.0223	.0195	.0213	.0190	.0202	.0163	.0196	.0188	.0155	.0151
	30	.0223	.0198	.0209	.0192	.0204	.0178	.0195	.0197	.0153	.0137
	60	.0222	.0185	.0208	.0197	.0191	.0164	.0191	.0187	.0151	.0139
20	90	.0222	.0214	.0204	.0204	.0210	.0207	.0240	.0221	.0161	.0142
	120	.0222	.0216	.0200	.0214	.0208	.0205	.0241	.0222	.0159	.0142
	150	.0221	.0219	.0220	.0210	.0206	.0203	.0244	.0228	.0169	.0147
	180	.0221	.0221	.0222	.0209	.0212	.0223	.0240	.0227	.0171	.0149
30	.	.	.	.	.	.	.	.	.	.	.
40	.	.	.	.	.	.	.	.	.	.	.
50	.	- .	.	.	.	.	.	.	.	.	.
60	.	.	.	.	.	.	.	.	.	.	.
70	.	.	.	.	.	.	.	.	.	.	.
	0	.1000	.1100	.1200	.1300	.1400	.1500	.1700	.2000	.3000	.4000
80	30	.1000	.1100	.1200	.1300	.1400	.1500	.1700	.2000	3000	.4000
	'	"	"	"	"	"	"	"	"	"	"
	180	.1000	.1100	.1200	.1300	.1400	.1500	.1700	.2000	.3000	.4000

Similar tables cover measurements at wavelengths:- 0.521, 0.606, 0.866, 1.243, 1.66 and 2.2 μm.

FIGURE 4.13 VARIATION OF ALBEDO WITH SOLAR ELEVATION ANGLE FOR A SAVANNAH SURFACE



Spectral albedos of a savannah for various wavelengths as indicated, given in micrometer, versus the solar elevation angle for low turbidity of the atmosphere (dashed lines) and for high turbidity (solid lines).

FIGURE 4.14 VARIATION OF ALBEDO WITH SOLAR ELEVATION ANGLE FOR A BOG SURFACE

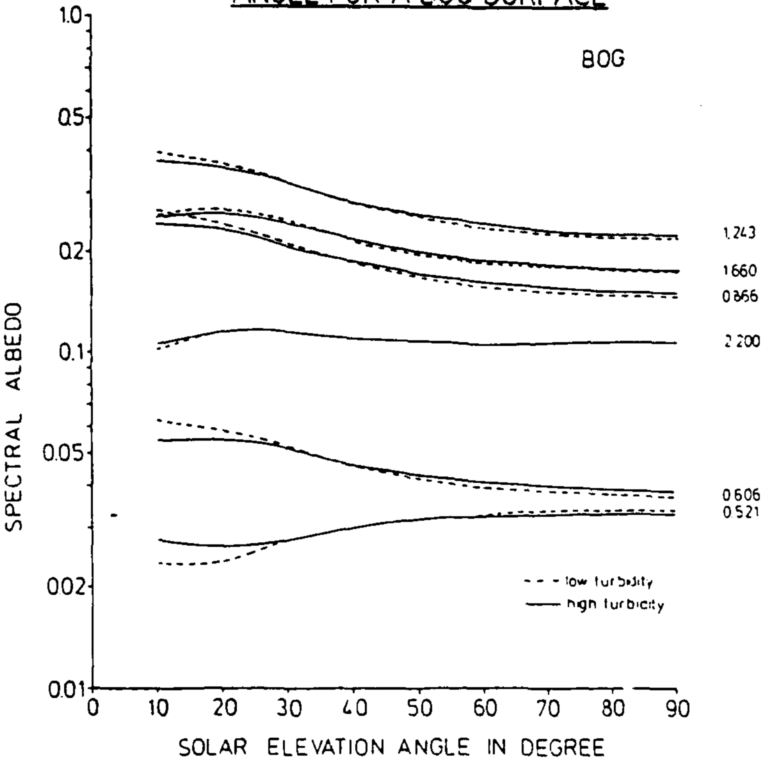


FIGURE 4.15 VARIATION OF ALBEDO WITH SOLAR ELEVATION ANGLE FOR A PASTURE SURFACE

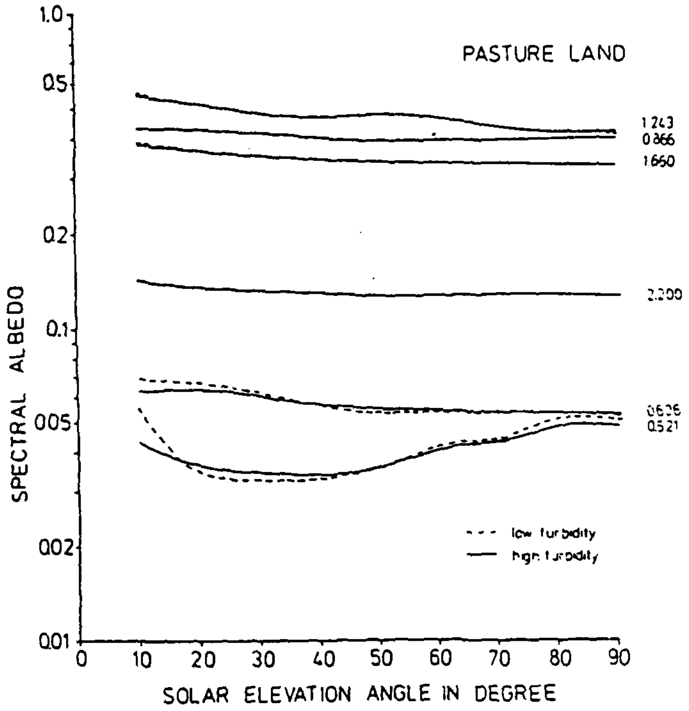
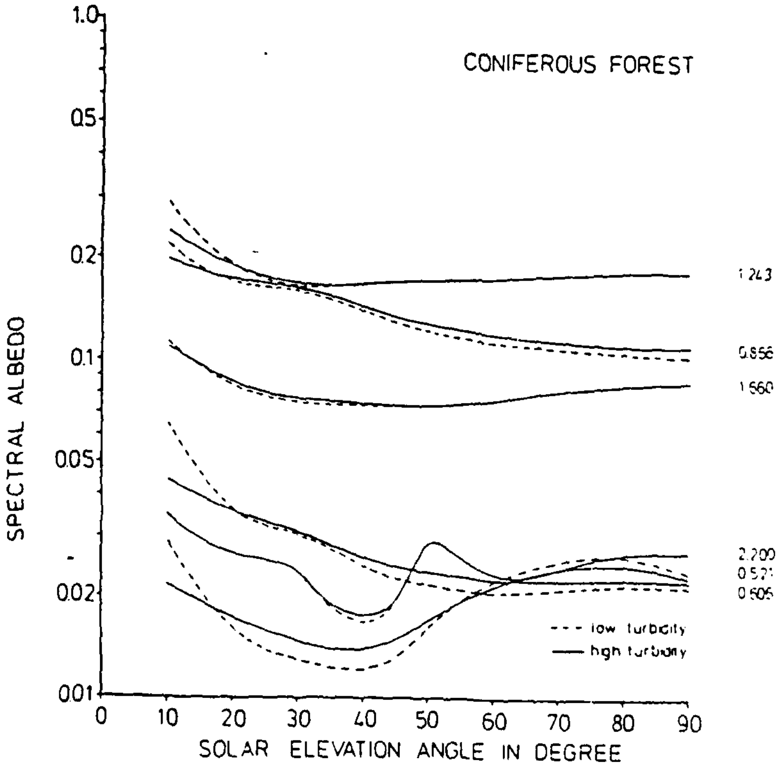


FIGURE 4.16 VARIATION OF ALBEDO WITH SOLAR ELEVATION ANGLE FOR A CONIFEROUS FOREST



(After Kriebel, 1977)

4.2.3.6 Sensitivity Analysis

The sensitivity analysis of RCI calculations to albedo was carried out in two parts:

(i) using a general albedo term α_g in both parts of **equation 2021 1g.2**,

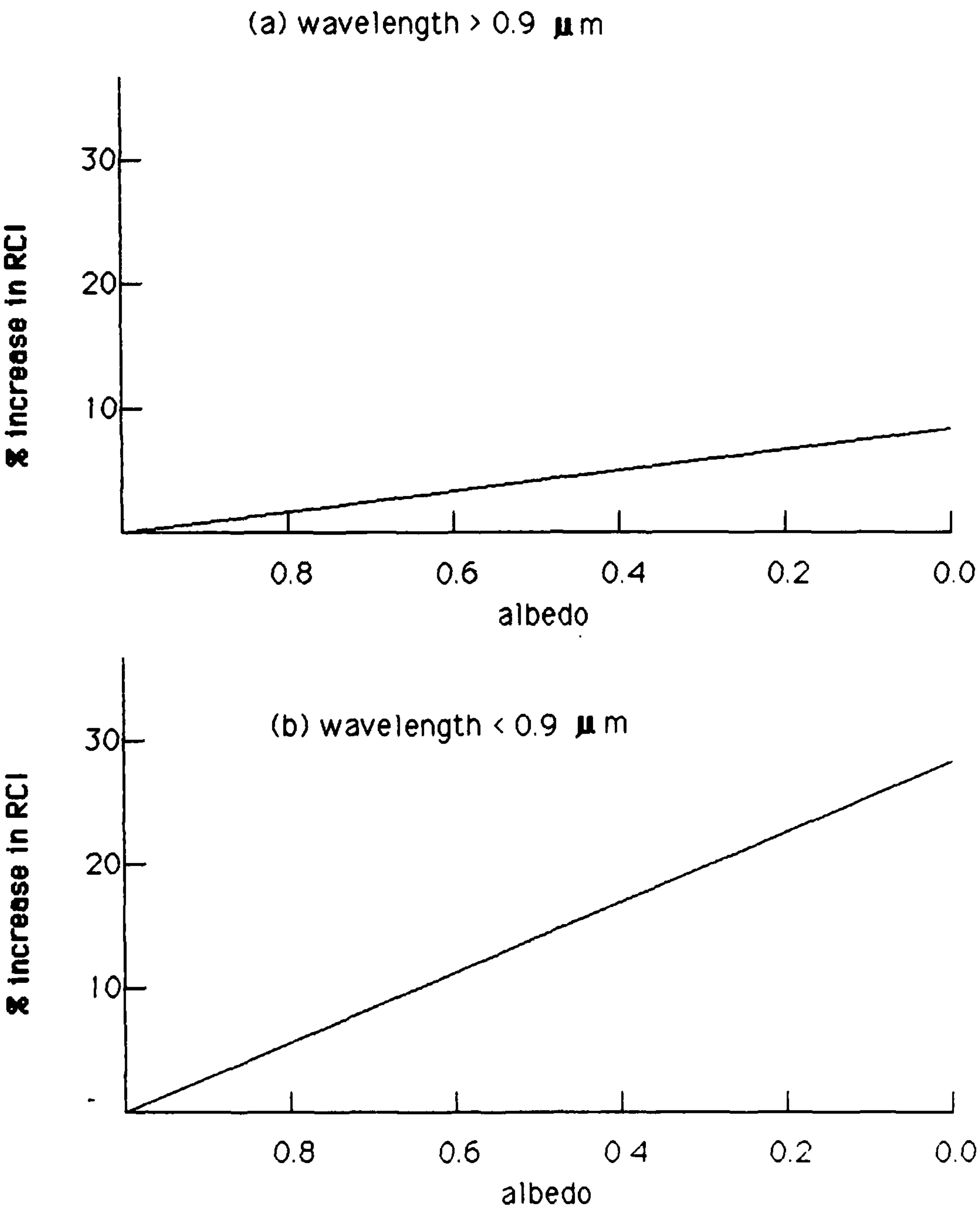
(ii) using two albedo terms α_{g1} and α_{g2}, i.e. albedo for wavelengths greater than 0.9 μm and less than 0.9 μm respectively.

Results of (i) are shown in **figure 4.7** where the influence of both different soil type and time of year are demonstrated. Considering the worst possible case, a 1% increase in albedo can result in a 0.3% decrease in RCI values. Using this information, the relative importance of those factors considered most influential on albedo variation (Henderson-Sellers, 1986) was determined (**section 4.2.3.5**)

The second sensitivity analysis using the two different albedo terms showed (**figure 4.17**) that the α_{g2} term is the most influential. The maximum possible influence of solar altitude and vegetation height are calculated in **table 4.3**. From these results, it can be seen that by using the two different albedos for their respective wavelengths, the total RCI variation possible is reduced from 43% to 26% which is a significant improvement. Thus **equation 2021 1g.2** can be stated as:-

$$S = (1-\alpha_{g1})\,[1-A(u^*,z)](0.349)S_0\cos z$$
$$+\,(1-\alpha_{g2})\,[(1-\alpha_0)/(1-\alpha_0\,\alpha_g)](0.651)S_0\cos z$$

FIGURE 4.17:RELATIONSHIP BETWEEN ALBEDO AND RCI



(a) wavelength > 0.9 μm

(b) wavelength < 0.9 μm

TABLE 4.3 Examples of Spectral Albedo for Different Vegetations
(From Kriebel, 1979)

BOG

Wavelength μm	Albedo	
0.35 - 0.49	-	
0.49 - 0.55	0.030	α_{g2}
0.55 - 0.68	0.048	
0.68 - 1.08	0.193	
1.08 - 1.45	0.290	
1.45 - 1.85	0.222	α_{g1}
1.85 - 2.65	0.113	

CONIFEROUS FOREST

Wavelength μ m	Albedo	
0.35 - 0.49	-	
0.49 - 0.55	0.018	α_{g2}
0.55 - 0.68	0.026	
0.68 - 1.08	0.145	
1.08 - 1.45	0.176	
1.45 - 1.85	0.080	α_{g1}
1.85 - 2.65	0.025	

PASTURE

Wavelength μ m	Albedo	
0.35 - 0.49	0.027	
0.49 - 0.55	0.039	α_{g2}
0.55 - 0.68	0.058	
0.68 - 1.08	0.414	
1.08 - 1.45	0.485	
1.45 - 1.85	0.353	α_{g1}
1.85 - 2.65	0.133	

4.2.4 Conclusion

This analysis has considered the worst possible case. By including the use of two albedos in **equation 20211g.2**, the total variation in RCI to possible errors in albedo estimation is 26%, i.e. +/- 13% around a central value. This is considered a marked improvement on the use of one spectral albedo.

This possible error will be further reduced when rain occurs, at different times of the year, and over different soil surfaces. It is proposed that when sufficient data exists for a 'typical' surface, that the version of **equation 20211g.2** stated in **section 4.2.3.5** be used to minimize the possible errors induced by variations in albedo. These variations in albedo can be difficult to quantify, therefore the original relationship will be used when only values of albedo integrated over the 0.5-2.2 μm spectrum are available. It is expected that the unaltered version of **equation 20211g.2** will be sufficient in the majority of cases.

4.3 User Friendly Techniques

4.3.1 Objectives

The main purpose of this project is 'to develop an operational system to predict off-road trafficability on a km^2 scale'. Within the context of this project the precise definition of the term 'operational' will provide some requirements for the system. The general requirements set out in **section 1.2** also require further definition (**section 4.3.2**). These requirements for the system divide into several techniques, such as, data bank development, menu driven system operation, coding design and system checking. These are discussed in the sections below.

4.3.2 Detailed Requirements

4.3.2a Construct a data bank facility containing information which will reduce the complexity of input data.

4.3.2b Design the code in a manner that facilitates replacement of subroutines; that identifies subroutines within the major design modules.

4.3.2c Design the system to be menu driven.

4.3.2d Design the system to carry out basic checks, such as, whether input data files exist, etc.

4.3.3 Development of Suitable Techniques

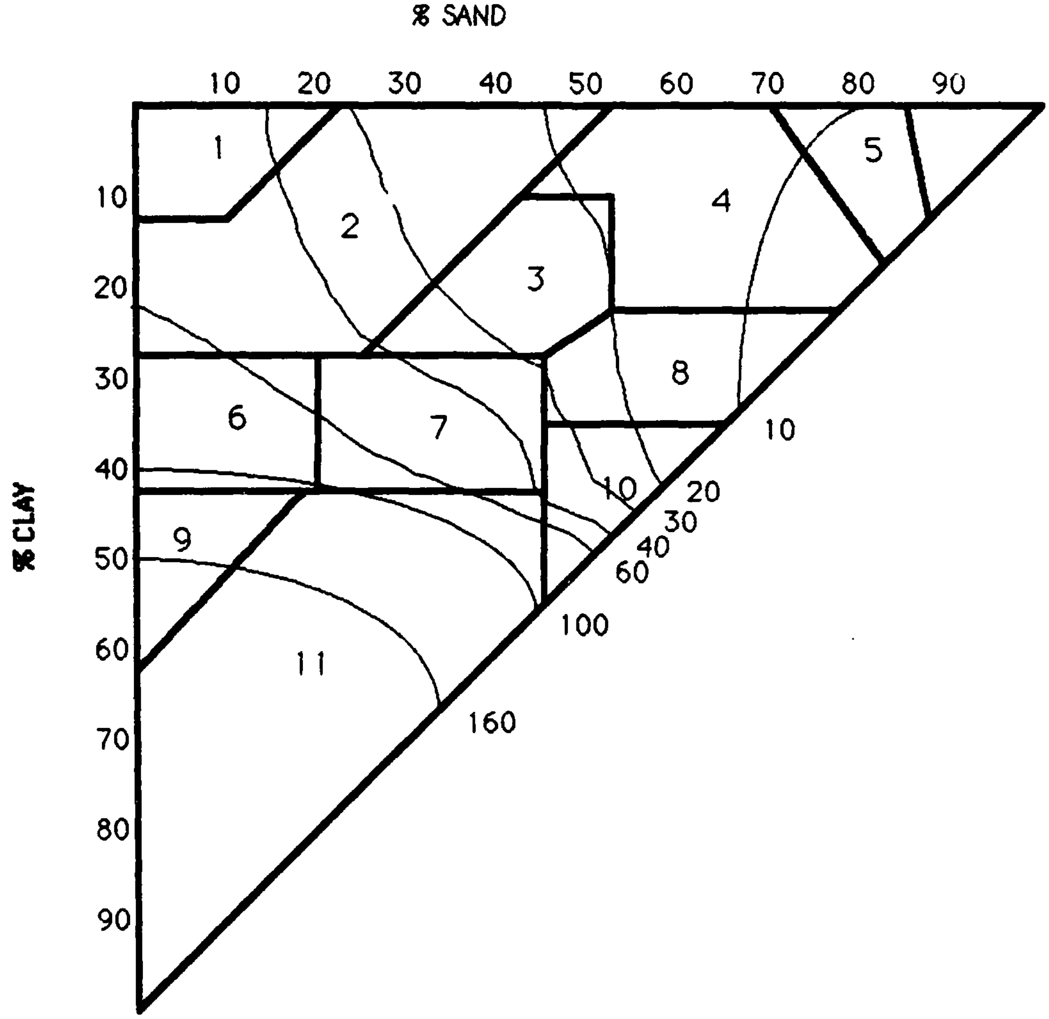
4.3.3.1 Data Bank Facility

In an effort of retain parsimonious data requirements a method of using categories of soil types was adopted. A choice of eleven different soil types is catered for (**figure 7.3**). The three soil layers (**figure 6.2**) may be defined by any of the eleven soil types. Through the specification of the soil type the system accesses a data bank (**algorithm 20211b, section 6.1.5**) to determine the following:- sand content (%), clay content (%), saturated water content, saturated hydraulic conductivity, mineral bulk density and bubbling pressure. These input parameters are required for the calculation of the soil moisture characteristic curves (**section 3.2**).

Research carried out by Brakensiek and Rawls (1983) produced a set of soil classification triangles which give ranges for the above parameters based on 5000 soil data sets. An example of the soil triangle for bubbling pressure is given in **figure 4.18**. The average and one standard deviation was identified and the the type of distribution determined. It should be noted that at this stage no allowance has been made for the effects of organic matter content or changes in porosity on these values. A value for each parameter is then calculated from the given distribution (**algorithm 20211b**) for use in the BSSS.

This is in effect an automatic 'look-up table' and substantially reduces the amount of work and decision making for the user. The use of a random selection from each parameter distribution includes a measure of the uncertainty and variability involved in trying to estimate these parameters.

FIGURE 4.18: EXAMPLE OF SOIL TRIANGLE SHOWING VALUES FOR BUBBLING PRESSURE (CMS) (FROM BRAKENSIEK AND RAWLS, 1983)



The develooment of this facility satisfies **requirement 4.3.2a**. The further development to include the effect of organic matter and porosity changes in this facility is further discussed in **section 10**.

4.3.1.2 System Design

The adoption of the strategy discussed in **section 1.4** enables the code to be written in a series of independent subroutines, any of which would be easy to replace with another and therefore satisfying **requirement 4.3.2b**. The naming of the subroutines has been carried out under the following rules:-

(**a**) Subroutines of the BSSS start with a 'b'.

(**b**) Subroutines of the VSAS2 start with a 'v'.

(**c**) Subroutines of the route evaluation scheme start with a 't'.

(**d**) Subroutines of the grid and optimum route scheme start with a 'r'.

(**e**) The remainder of the name, after the initial letter, should be something meaningful, e.g. 'bcntl' refers to the BSSS control subroutine, 'vlatflo' refers to the calculation of lateral flow in the VSAS2, and of course 'link' links all the different schemes together to form the BORTS.

By adopting this naming strategy the further development and maintenance of the system should be much easier. This is considered an important but often neglected facet of user friendliness.

4.3.3.3 Menu Driven Systems and Basic System Checks

The running of the BORTS is controlled by a series of menu driven decisions made by the user. **Section 9** gives examples of how to run the various schemes within the BORTS. The information given in plain text is what the user will see on the screen, the user is expected to make choices (outlined text) or input names of files as directed by the system.

The facility also allows the user to edit input files by category and the specific name. This system provides the existing value of the variable or parameter and asks for a new value. This is considered a vast improvement in that the user is able to interrogate the input files through specific variable names rather than having to identify its position within a data file. This eliminates the user having to deal with all the various formats that may exist in the input data files. This facility has also been added to the VSAS2 as the input for this system is much more complex than that of the BSSS.

The basic system checks carried out include checking if an input file exists though it is unable to check whether the user has specified the correct one. When the system asks for an output file and the user returns one that already exists, a message will ask the user to confirm that they wish to overwrite that existing file. Out of range replies to the menus are also catered for in that the user will be asked to re-enter their reply.

4.4 Conclusions

This section has dealt with the improved BSSS model components. It should also be noted that the development of the soil strength relationship is not limited in application to the BSSS, but has also been added to the VSAS2. Also, the development of a menu driven system making basic checks has been added to the VSAS2 and the subroutines have been renamed from those of Whitelaw's (1987).

The development of an initial strategy to deal with the problem of albedo within the non-isothermal evaporation routine is considered an important step forward and will be great interest in further development and evaporation studies.

The increased user friendliness of the scheme for both the user and developer is a major step towards the prime objective. With the techniques discussed in **section 4.3.3.3** applied to the route management scheme (**section 5**) the BORTS will have included aspects of content, application and utility and therefore a well balanced system.

5 ROUTE SELECTION PROCEDURES

5.1 Introduction

The development of the BSSS and the adaptation of the VSAS2 enable the prediction of soil strength and hence off-road trafficability with variable temporal resolution. The BSSS simulates the soil water physics of a column, while the VSAS2 simulates the soil water physics of a sloping segment. The two systems have both advantages and drawbacks over each other (**section 8**), but as yet the spatial resolution of the schemes has not been discussed. This section sets out the requirements and methodologies developed to include spatial resolution and apply the RCI predictions to route problems.

5.2 Representation of Spatial Resolution

The initial problem of trying to model the real world is to simplify its complex continuous nature into something that is identifiable and discrete. That is, to represent the real world by something that can be handled by the 'viewer'. The 'viewer' may be a computer system, software, a specialist or a trained user and the degree of simplification may often be dependent on all these 'viewers'.

Within the BORTS, two independent schemes exist to predict off-road trafficability, namely the BSSS and the VSAS2, the utility of which are discussed in **section 7**. Each run of the BSSS has a set of clearly defined input information and can therefore be said to represent all areas which can be described by that information. When that information changes, i e different soil type, different meteorological or site conditions, the BSSS is run with the appropriate input information.

To apply this concept of spatial resolution to an area, a clearly defined strategy of simplifying the real environment must be formulated. This strategy must include the restrictions imposed by the hardware, software and the user. **Section 7.5** sets out the guidelines for the user on how to set up a grid system over an area. The important site characteristics affecting the set-up of a grid are those of soil type and topography (i.e. initial moisture contents and biased hydraulic conductivities), but the user may at their discretion also include the influence of a moving storm across an area in the meteorological information.

Having decided how many BSSS runs can be made with the required computer time (**section 7.5**) and consequently the grid format to be used to represent the area, the user can use the VSAS2 to simulate the processes taking place in one or more of the grid squares (**figure 6.11**). The increase in computer resources will have to be assessed by the user (**section 10**).

The implication of being able to represent an area of particular interest will be of great importance to route evaluation or selection problems which must pass through that particular grid square. The price to pay for running the VSAS2 scheme as opposed to the BSSS will be in the increase in both computer time (**section 10**) and in user time for preparation of the input information (**sections 6.2** and **7.3**).

5.3 Calculation of Optimum Route

The application of the RCI preaıctions along with their representation over an area defined by the grid system discussed above, allows optimum routes to be determined. The optimum route must combine the effects of soil strength (RCI), speed of travel and also shortest distance. The calculation of the optimum route from a specified start position to a specified finish position will be dependent on the following factors:-

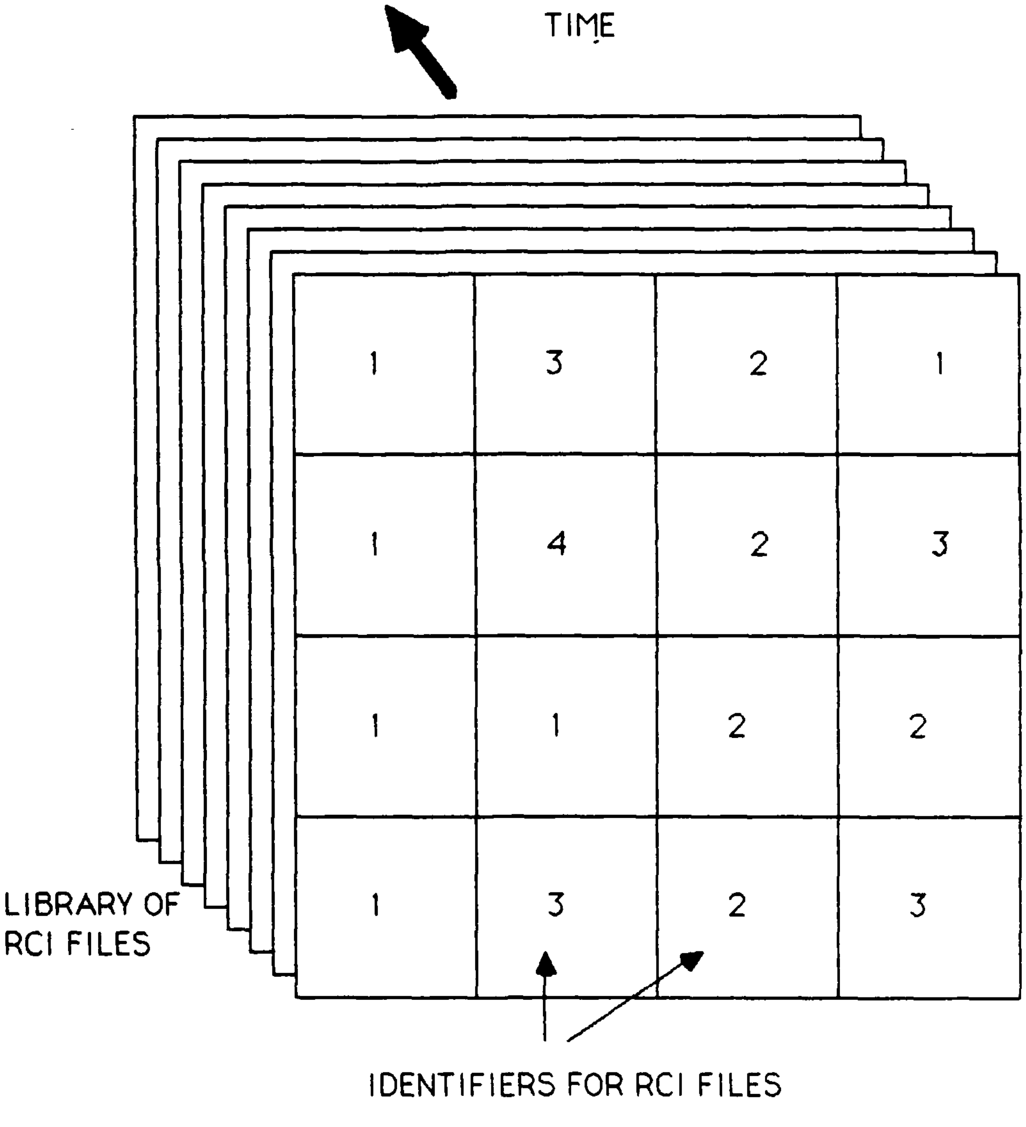
(a) Time of Journey Start

Throughout the duration of a storm, the status of the soil surface layers (**figure 4.1**) will alter according to how much water is added to it and its initial moisture conditions. The critical layers as shown in **figure 4.1**, will be subjected to drainage and may be affected by evaporation. The RCI over the area will therefore be liable to fluctuations mirrorring these processes. Hence, the start time of any journey may affect the calculation of the optimum route from start to finish.

Using the predicted RCI library (**figure 5.1**) over the area, it would be possible to calculate the optimum route at each time interval that there are RCI values for. This could be linked in with the RCI restricted travel speed of the vehicle over the route and the probability of becoming bogged down. This would be a significant aid in decision making for not only the best route to take, but also the best time to start that journey. An example of this application is shown in **figure 5.2**:-

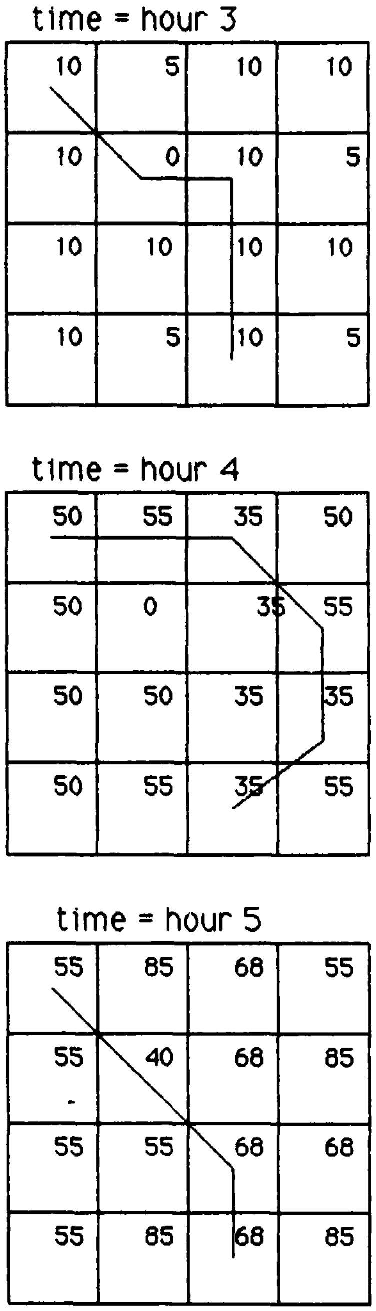
(i) Scenario A shows 'time = hour 3' when the vehicle would only have a negligible chance of completing any route without bogging down.

FIGURE 5.1 LIBRARY OF RCI OVER A GRID



1	3	2	1
1	4	2	3
1	1	2	2
1	3	2	3

FIGURE 5.2 EFFECT OF RCI ON JOURNEY START TIME



time = hour 3

10	5	10	10
10	0	10	5
10	10	10	10
10	5	10	5

Scenario A

Vehicle will only have a negligible chance of completing any route without bogging down

time = hour 4

50	55	35	50
50	0	35	55
50	50	35	35
50	55	35	55

Scenario B

Vehicle could complete route shown in three hours

time = hour 5

55	85	68	55
55	40	68	85
55	55	68	68
55	85	68	85

Scenario C

Vehicle could complete route shown in one and a half hours

(ii) Scenario B shows 'time = hour 4' when the vehicle could complete the calculated route in 3 hours.

(iii) Scenario C shows 'time = hour 5' when another route becomes feasible with a travel time of 1 1/2 hours.

By waiting until 'time = hour 5' a small area which was previously unpassable becomes able to allow the passage of the vehicle and so allows the route in scenario C to be used. The vehicle completing scenario B will arrive at its destination at 'time = hour 7', whereas the vehicle completing will arrive at 'time = hour 6.5'. That is, by waiting an hour scenario C becomes not only the faster route but will save on total travel time and hence fuel resources.

(b) Approximate Distances Involved

As the vehicle travels across the area, the environment is continually changing and hence the state of the soil may also be changing. The distance of the route combined with the speed of the vehicle become increasingly important as distance increases and speed decreases.

For a given speed, as the distance increases so does the likelihood of the RCI for a particular sub-area having changed from 'time = start time'. It is at this stage that the area represented by each grid square and the output interval of RCI values must be considered. For example, say a library of RCI values (**figure 5.1**) has been created at hourly intervals for a grid with grid squares of approximately 5 km^2. Ignoring the effect of RCI on the speed of the vehicle, assume that the vehicle is moving at a steady speed of 20 kms hr^{-1}. It therefore follows that if the journey starts at 'time = hour 1', that after the vehicle has travelled 20 kms or approximately 4 grid squares the RCI values over the area are no longer be represented by 'time = hour 1' but by 'time = hour 2' and should accordingly be changed throughout the duration of the journey.

(c) Speed of Vehicle

As discussed above, an important facet of route management is including the effect of changing RCI values throughout the duration of a journey. The point at which the next set of RCI values predicted for the grid should be implemented depends on output interval, distance covered and speed of the vehicle. In a first attempt to include these effects an average speed across the area is assumed and the point at which the RCI library is accessed to update the status of RCI across the grid always happens at grid square boundaries. The restrictions of these assumptions are further discussed in **section 8**. Initial results from the Ministry of Defence's DRIVEB speed prediction model, which is dependent on RCI values, are discussed in **section 10** with respect to the improvement this facility would bring to a second generation route management scheme.

(d) RCI Values Across the Area

The concepts above have been based on the existence of a library of RCI values (**figure 5.1**) which can be both created at different output intervals and accessed at any output time. The calculation of an optimum route with a known start and finish grid square positions will be based on the shortest distance and the largest RCI values. From any grid square position the RCI values of the 3 grid squares in the major direction of movement are biased according to the shortest distance. The grid square with the largest RCI value will normally be chosen (see below for exceptions). This will have important implications for application to route speed analysis.

(e) Errors in Logic - Looping and Dangling

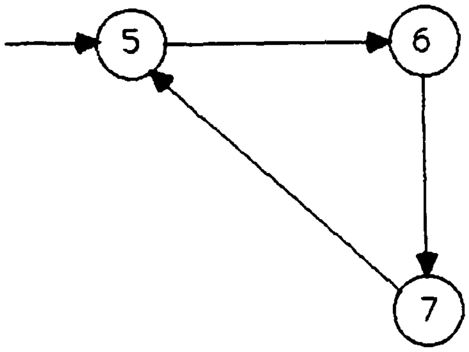
In the calculation of an optimum route there may occur two errors in logic:-

(i) Looping:- **Figure 5.3a** shows the situation of looping. The situation can be described by "move 7 depends on move 6 which depends on move 5 which depends on move 7 which depends on move 6......". A check must be included in the optimum route calculation to ensure that this situation is not allowed to occur.

(ii) Dangling:- **Figure 5.3b** shows the situation of dangling. The situation shows that move 6 should be to grid square A but that this move would leave the route unable to continue. The system would return that a route is not possible under this set of conditions. The correct decision would be for move 6 to be the next best grid square, i.e square B. There is a problem in allowing for this type of logical error in that, there are special occasions when 'dangling' can be accepted. These will include when it happens at a specified start grid square and when it happens at the required finish grid square.

FIGURE 5.3 ERRORS IN LOGIC

(a) Looping



"move 7 depends on move 6 which depends on move 5 which depends on move 7 which depends on move 6 ...."

(b) Dangling

40	45	35	35
55	55 M5	60 B	70
0	100 A	0 C	60
0	0	0	50 F

key:

F - Finish square

M5 - Move number 5

A - Best square for move 6 without considering next move - dangling would occur, i.e. all options are 'no-go' at A

B - Next best move to stop 'dangling' at A

C - Third square considered at move 5 but has a 'no-go' status

5.4 Application to Problem Routes

5.4.1 Calculation of Routes

The situation may arise where not only does a route have specified start and finish grid squares, but might also have one or more specified intermediate grid squares the route must go through. This type of problem can be treated as a series of smaller routes where the intermediate way points are considered as temporary finish positions.

These way points or intermediate grid squares may be represented by:-

(a) a BSSS run,
(b) a VSAS2 run, or
(c) a sub-grid system comprised of any combination of BSSS and/or VSAS2 runs.

The representation of a grid square by a sub-grid system could have important implications for restricted routes as sub-routes within the one grid square can be calculated. The sub-grid system could also be used to calculate minimum RCI over the area or the area of the square which has RCI values below some predetermined threshold value.

5.4.2 Evaluation of a Route

The evaluation of RCI along a specified route will have important application to speed prediction. This will be important when trying to compare several different routes. Different routes will have different recovery characteristics when the scenarios discussed in **section 5.3(a)** may be of particular application.

The evaluation of a route does not require a full grid set-up, and a simple example is shown in **figure 6.17a**. The user has only to describe the route as a series of nodes having a distance along the route, an identifying number relating to the library of RCI files and a time that the vehicle will approximately reach that node.

5.5 Conclusions

This section has outlined a framework for applying the BSSS and VSAS2 off-road trafficability schemes. The development of a grid system which may be part of a hierarchy of grid systems to represent the real environment, enables the inclusion of spatial resolution of site and meteorological parameters and variables.

The use of a grid system of course simplifies the real environment and consequently incurs inaccuracies in prediction. But, there is great potential for the user to build up the grid sophistication as more information is acquired and as the user becomes more skilled and familiar with both the system and the area under consideration.

The application of the results from the BSSS and the VSAS2 over an area to route management problems is considered an important aspect of the development of the BORTS. Development of the route management and grid set-up scheme (**sections 6.4** and **6.3** respectively) enables the user to utilise the RCI information calculated in a manner that does not require the users assessment of many numbers.

It can therefore be said that the route management scheme is a user friendly facility along with those discussed in **section 4.3**. It allows the non-expert to utilise information that normally only the experienced could analyse. The route management scheme also displays the potential use of the BSSS and VSAS2 schemes and as discussed in **section 1.1**, this application could easily be extended to agricultural applications.

6 PROGRAM DESIGN

6.1 DESIGN: Bristol soil strength scheme

In a first attempt to solve the requirements set out in **section 1.2**, a basic infiltration and evaporation scheme devised by Anderson, 1982, was used as a starting point for further development. This development included devising a suitable soil moisture - soil strength scheme, a first attempt to include some effects of vegetation on soil strength and evaporation calculations, a first attempt to include the effect of topography on initial moisture contents and the 'biasing' of hydraulic conductivities to simulate 'flat top', 'slope' and 'flat bottom' areas. This is shown in **figure 6.1** alongside the existing empirical scheme, SMSP.

6.1.1 Specification of Detailed Requirements

As discussed in **section 3.1**, there are to be two methods of predicting soil moisture and hence soil strength. This section will specify the requirements for the BSSS:-

6.1.1a An area is represented by a one dimensional soil profile, such as that shown in **figure 6.2**. There can be three potentially different layers and the total number of cells in the profile may be up to ten. The cells are proportioned between layers by the user.

6.1.1b Soil water modelling within the soil profile should be as deterministic in nature as possible, chosen from the available

FIGURE 6.1 BSSS versus SMSP

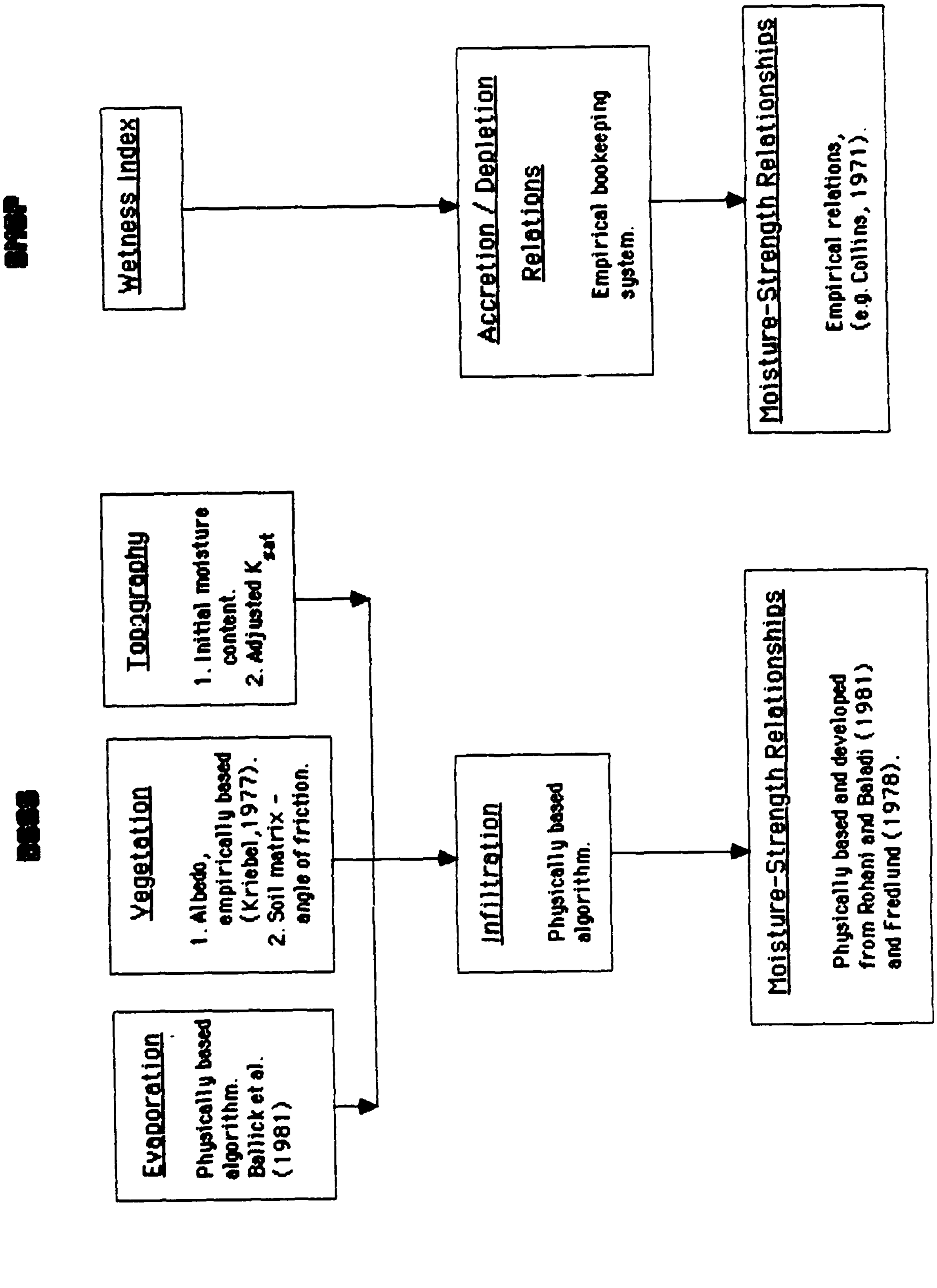
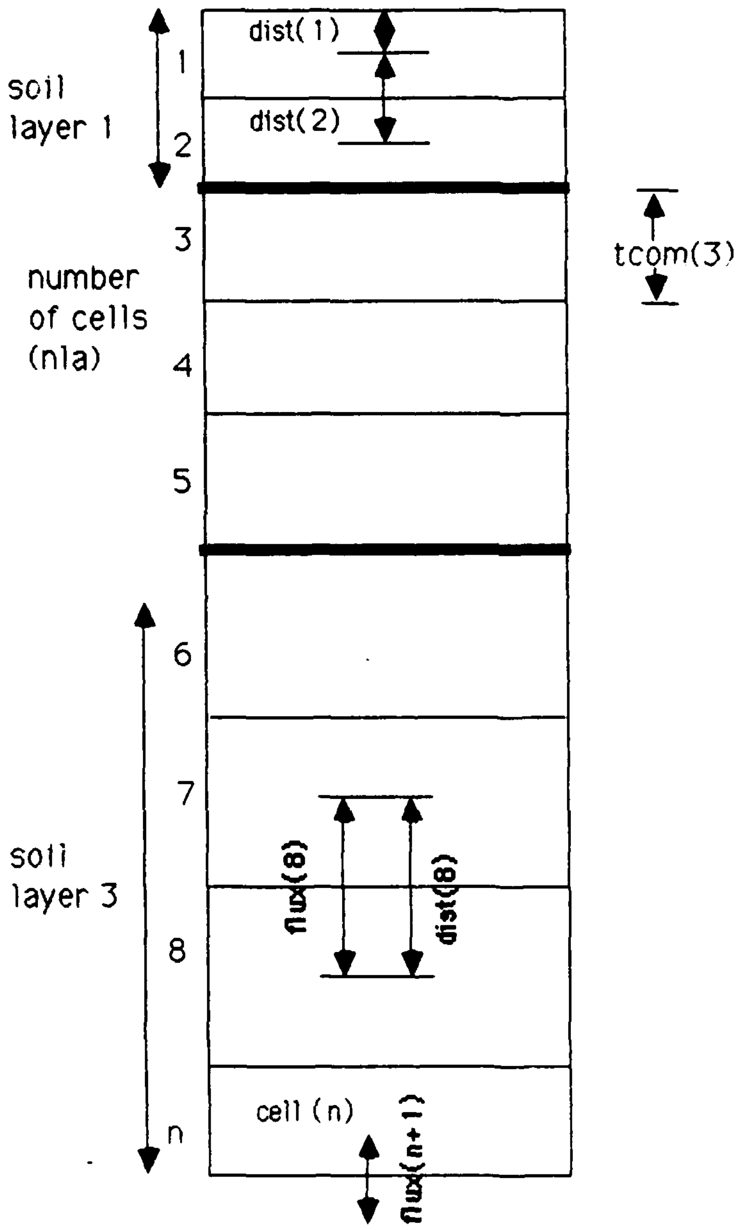

SMSP
Wetness Index
Accretion / Depletion Relations
Empirical bookkeeping system.
Moisture-Strength Relationships
Empirical relations, (e.g. Collins, 1971).
BSSS
Evaporation
Physically based algorithm. Ballick et al. (1981)
Vegetation
1. Albedo, empirically based (Kriebel,1977).
2. Soil matrix - angle of friction.
Topography
1. Initial moisture content.
2. Adjusted K_{sat}
Infiltration
Physically based algorithm.
Moisture-Strength Relationships
Physically based and developed from Rohani and Baladi (1981) and Fredlund (1978).

FIGURE 6.2 : ONE DIMENSIONAL SOIL PROFILE



relationships (**section 3.2**). The calculation of soil moisture, soil water potential and hydraulic conductivity of any cell at any time should be an inherent feature of the model.

6.1.1c Evaporation must be calculable at the same temporal resolution as the infiltration routine. It should demonstrate the non-isothermal nature of evaporation throughout the day on any day of the year in a temperate latitude (**section 3.4**). The influence of site orientation and other characteristics should be considered.

6.1.1d The scheme should be computationally quick to run (**section 10.2**).

6.1.1e Soil strength must be calculable from readily available soil parameters and include the effect of soil water (**section 4.1**).

6.1.1f User input must be readily available or calculable from readily available data. Complex soil parameters may be stored in a databank facility according to soil classification (**section 4.3**).

6.1.1g The scheme should be user friendly (**section 4.4**) in terms of:-

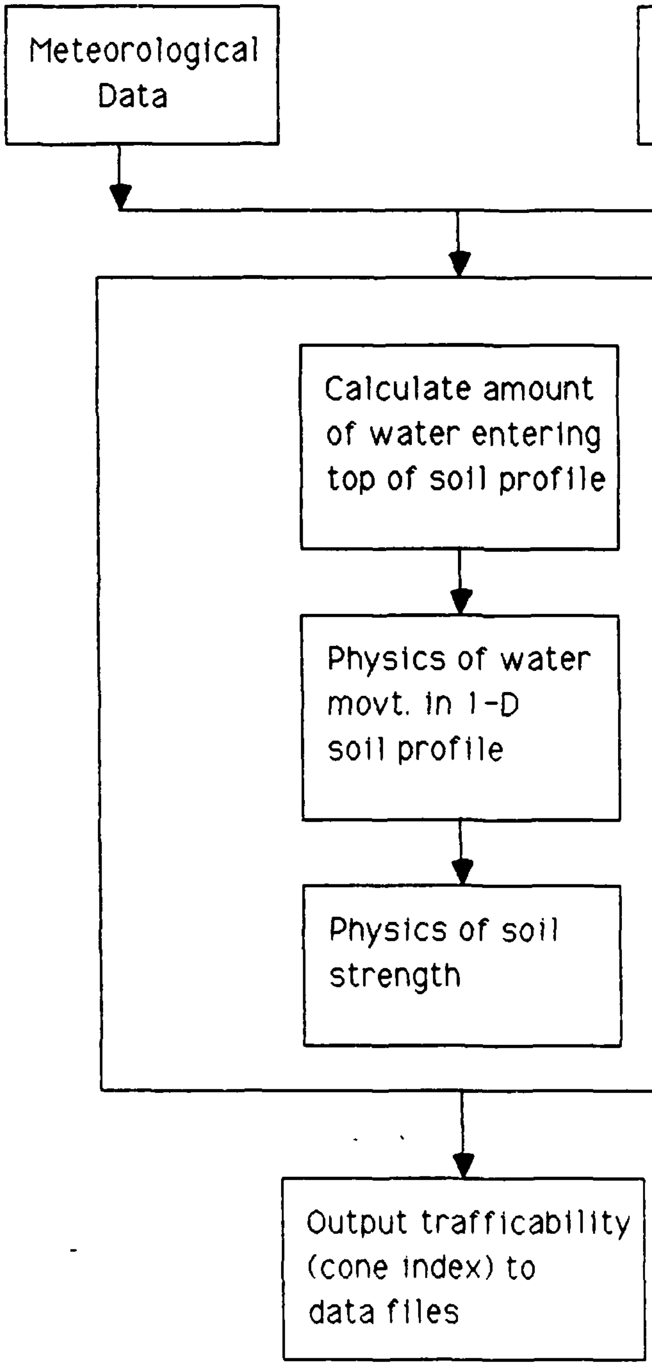
(i) system menu information when choices are to be made,

(ii) system prompts for user response,

(iii) system checking for 'basic' errors - such as whether files exist or if the user really means to overwrite an existing file.

This clarification of the problem (stage one, **figure 1.2**) led to the design of a solution (stage two, **figure 1.2**) and is shown in **figure 6.3**. The data requirements of the scheme divide into (1) the site specifications

FIGURE 6.3: BSSS - SYSTEM DESIGN


Meteorological Data
Site Specifications
Calculate amount of water entering top of soil profile
Physics of water movt. in 1-D soil profile
Physics of soil strength
Output trafficability (cone index) to data files

which are mainly static in nature, and (2) the meteorological data which is dynamic in nature. These parameters and variables are manipulated by the main body of the scheme to calculate soil water movement, evaporation and hence soil strength on a variable temporal scale. Output from the scheme is for use in the route management scheme.

Further refinement of the system design and choice of methodologies to implement the solution (stage three, **figure 1.2**) allow the program design to be constructed as in **figure 6.4**. The algorithms and data structures to be used are specified in **section 6.1.5** and the code formulated (**Appendix B**) is described in **sections 6.1.3** and **6.1.4**.

6.1.2 Development of System Design

The system design, as described in **section 1.4.2**, tries to specify a solution to the above requirements. The general system design of the BSSS is shown in **figure 6.3** and is comprised of four primary modules. Three of these primary modules deal with input - output requirements, while the remaining module (known as the functional module) has been subdivided into three. This top -down approach, called modularity, allows the developer to consider the problem as a whole and hide details that may be distracting. The developer can access lower levels for more detail when required.

At this stage of development the ideal number of modules is about 5-6. More than this and it becomes very difficult to understand what the system is to do, trace the flow of data through the system, and identify areas of complexity. It should be remembered that this does not make the

FIGURE 6.4 : BSSS - PROGRAM DESIGN

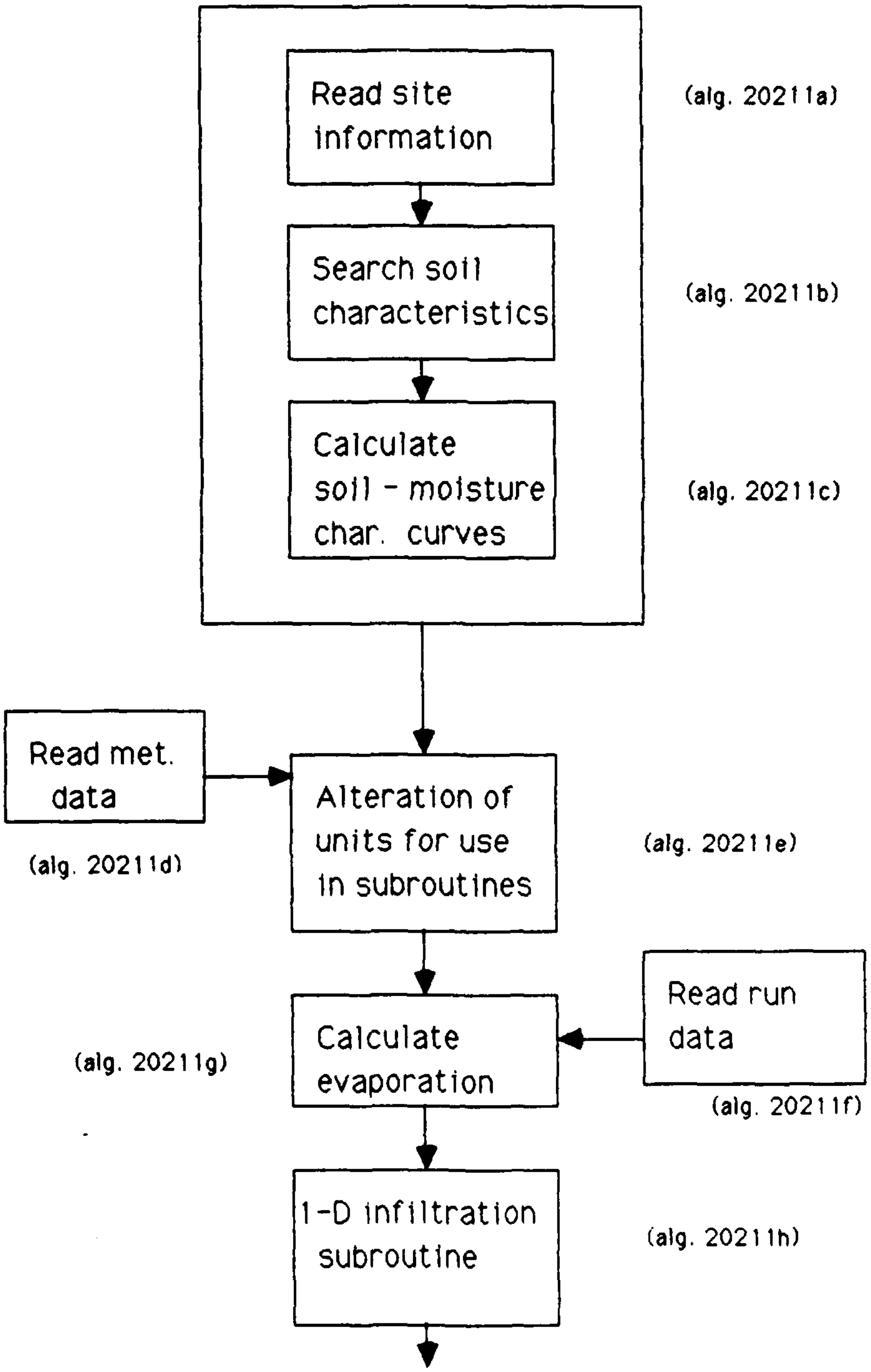
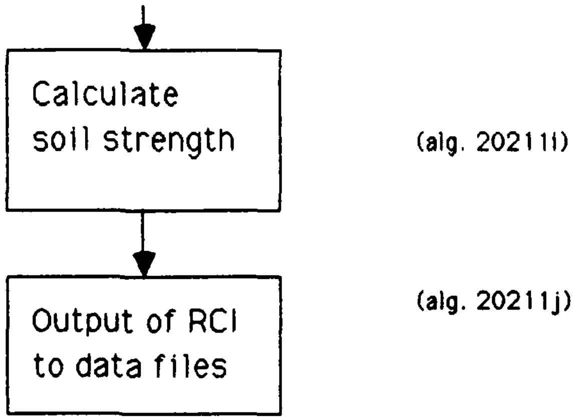


FIGURE 6.4 (cont.) : BSSS PROGRAM DESIGN



problem any easier, but it does allow for the identification of the the areas that will be most difficult to handle. Hence, the chance of confusion is greatly reduced and there is less chance of entanglement with unrelated data and functions.

The system design for the BSSS (**figure 6.3**) identifies that there are two major categories of data requirements for the system. These have been divided by type into static and dynamic, i.e., site specifications and meteorological data respectively. The information input to the system is then converted into predictions of trafficability for output. The 'functional module' transforms the input data by evaluating the amount of water (**section 6.1.1c**) entering the soil profile described in **section 6.1.1a**, simulating the flow of water through the profile as described in **section 6.1.1b**, and finally calculating the soil strength as defined in **section 6.1.1e**. Further refinement of the system design into the program is described in the following section.

6.1.3 Description of Program Design

The modularity of the system design is carried on through to the program design as can be seen in **figure 6.4**. Here the modules have been further refined into smaller sub-modules. Each component of the program design is now described in detailed pseudo code (**section 6.1.5**) for the purpose of constructing code (**Appendix B**).

The input requirements have now been divided into three categories:-

(a) site information, i.e. site characteristics that the one soil profile will represent and is fully described in **algorithm 20211a**,

(b) meteorological data used in the evaporation calculation and determining how much water is available to infiltrate the soil (**algorithm 20211d**), and

(c) run specification information as described by **algorithm 20211f**.

The program design shown in **figure 6.4** indicates when this input data is required by the system. The system searches an established data bank (**algorithm 20211b**) for more information about different soil types. From this information the soil moisture characteristic curves as defined in **section 3.2** are calculated (**algorithm 20211c**). Some of the meteorological information is input in units that require alteration and these are defined in **algorithm 20211e**.

The following sequence is then carried out at the iteration time intervals for the duration of the run period. The amount of evaporation is calculated (**algorithm 20211e**) as described in **section 3.4**. The amount of water entering the soil profile is the calculated and the infiltration and redistribution of soil water within the soil profile is calculated according to Richard's equation as described in **section 3.2.4** and by algorithm **20211h** in **section 6.1.5**.

The soil strength is then calculated for the top soil cell according to **algorithm 20211i** and as developed in **section 4.1**. The RCI is then written to the data files (**algorithm 20211j**) for use by the route management scheme (**section 5**).

The program design is an intermediate stage between the English description and the program code and is vital for the further development of the scheme. The relation of the code with the design is discussed in the following section.

6.1.4 Program Implementation

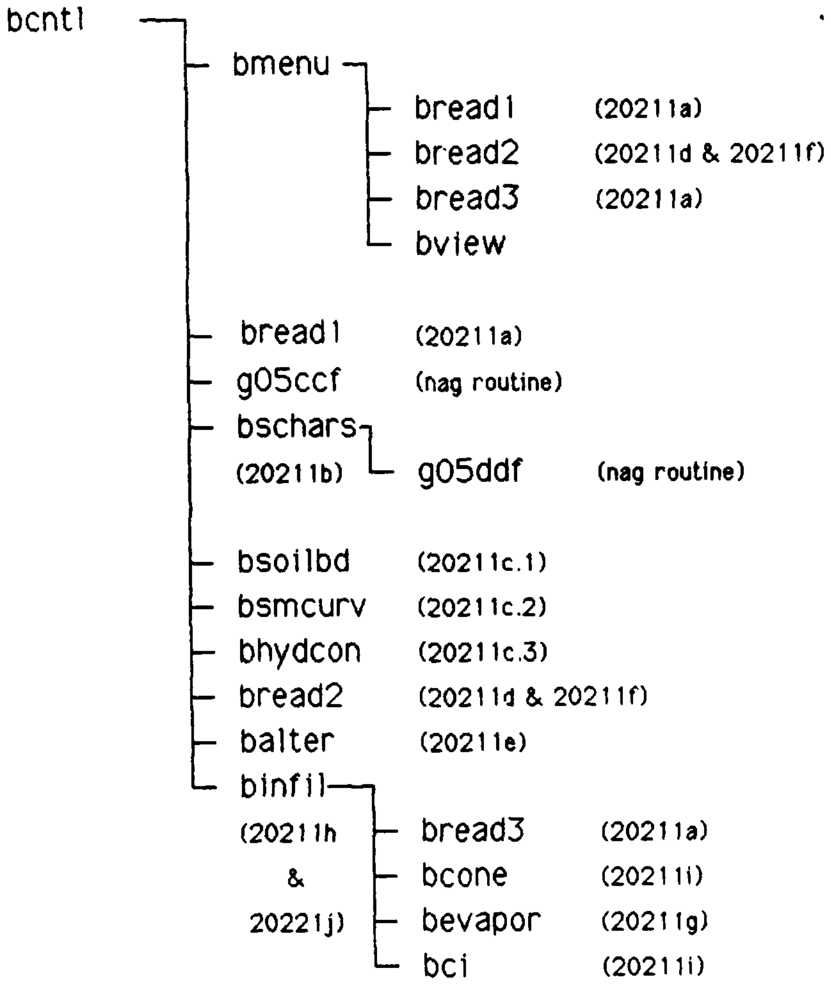
Following the program design and algorithm specifications given in **section 6.1.5**, FORTRAN code (**Appendix B**) was constructed to solve the algorithms. As in the other modules (**figure 1.3**), the consistency in programming style has been maintained through the construction of small separate subroutines called by a controlling subroutine. This programming style allows the development, replacement or addition of further subroutines with ease.

The following section will link the FORTRAN code (**Appendix B**) with the appropriate algorithms (**section 6.1.5**) and how they work together to solve the requirements set out in **sections 1.2** and **6.1.1**.

The 'subroutine tree' shown in **figure 6.5** demonstrates the flow of control within the BSSS. The appropriate algorithms coded in the subroutines are given in brackets. All subroutine names used in the BSSS start with a 'b' for ease of identification from the other fourth level module subroutines. The other module subroutines also have a similar system of identification.

The flow of the BSSS is controlled through one major subroutine which has been called 'bcntl' and can be considered as representing module 2022 in **figure 1.3**. It is worth noticing that those subroutines in BORTS which control the flow of activity are named '_cntl' where the underscore is an identifying letter - namely 'b' for the BSSS, 'v' for the VSAS2, 'r' for the optimum route management scheme, and 't' for the route evaluation scheme.

FIGURE 6.5 : BSSS SUBROUTINE TREE



The first decision made is whether to edit a BSSS input file (module 20212), i.e. call 'bmenu', or whether to run the BSSS (module 20211), i.e. bypass 'bmenu'. Subroutine 'bcntl' has been developed to allow the user to run up to thirty files at any one time. The names of the input files is requested by the system and an 'identification' integer is asked for to identify the output file 'data.n', where n=1,2....30 . The default 'n' value starts at one, but the system allows the user to interactively change this. Subsequent values of 'n' are automatically incremented, though the user may change these as well.

For each input file and associated output file the BSSS is run. Firstly the input data is read via 'bread'. Subroutine 'bread' has three entry point - i.e. 'bread1', 'bread2' and 'bread3'. This methodology facilitates future development. The input is used to access the data facility detailed in algorithm 20211b, and is coded in subroutine 'bschars'. Subroutine 'bschars' calculates soil properties from a NAG routine applied to stored distributions in 'spec.bd1' and 'spec.bdata'. The soil properties are calculated for the three soil layers and each layer may be allocated a different soil type according to the classification given in **figure 6.6**. The user also provides information on the organic content of each layer and then the soil bulk density is calculated in 'bsoilbd' and passed to 'bsmcurv' where the suction - moisture curve for each soil layer is calculated. To complete the soil - moisture characterization for each soil layer, 'bhydcon' is called to calculate the soil moisture - hydraulic conductivity curves.

Subroutine 'bcntl' then reads more input data so that the user friendly units that some meteorological information are given in, can be suitably altered in 'balter'. The remaining soil strength variables are read within

FIGURE 6.6 : SOIL CLASSIFICATION

INDEX	SOIL TYPE
1	Silt
2	Silty loam
3	Loam
4	Sandy loam
5	Loamy sand
6	Silty clay loam
7	Clay loam
8	Sandy clay loam
9	Silty clay
10	Sandy clay
11	Clay

'binfil' by 'bread3', and the static section of the soil strength relationship given in algorithm 20211i is calculated before the simulation of the flow of water into and through the profile during the run period. Subroutine 'binfil' calculates the infiltration and re-distribution of water within the profile according to algorithm 20211h (Richard's equation) and calls 'bevapor' to calculate evaporation from the surface according to the non-isothermal routine detailed in algorithm 20211g. Matric suction values are passed from 'binfil' to 'bci' where the dynamic soil strength is calculated in terms of rating cone index (RCI). Output of RCI through time according to algorithm 20211j is controlled from within 'binfil'.

Control is returned to 'binfil' at the end of the run period. The input files and output files are closed and the program checks whether another file is to run. When no more files are to be run, the user is returned to the master menu where the user may run any of the modules shown in **figure 1.3**.

To facilitate further development of the BSSS, **Appendix A** describes the variable, parameter and counter names used in the scheme. Each name is accompanied by a brief description of their use and any other subroutine they are passed to or from and the name of the common block they can be found in.

6.1.5 Specification of algorithms

This section sets out the algorithms used in the BSSS. The solutions are written in 'pseudo-code', which is an intermediate stage between the description given in English and the program FORTRAN code. This section is represented as stage 3, the program design, shown in **figure 1.2**.

6.1.5a List of algorithms in the following section

Algorithm 20211a :Read site information.

Algorithm 20211b :Search soil characteristics.

Algorithm 20211c :Calculate soil moisture characteristic curves.

Algorithm 20211d :Read meteorological data.

Algorithm 20211e :Alteration of units.

Algorithm 20211f :Read run data.

Algorithm 20211g :Non-isothermal evaporation routine.

Algorithm 20211h :One dimension infiltration.

Algorithm 20211i :Calculate soil strength.

Algorithm 20211j :Management of output data.

ALGORITHM 20211a : READ SITE INFORMATION (BSSS)

DEFINE

(a) data file to be read

(b) output file identifier   interactive system prompted

(i.e. data.1,data.2,.....data.n)

READ from input data file the following information:

1. internal angle of friction	(phi)
2. remoulded angle of friction	(phib)
3. soil cohesion	(c)
4. soil shear modulus	(gg)
5. depth of cone tip	(zz)
6. number of soil layers	(nos)
7. soil type of layers	(isty(1..nos))
8. organic content of layers	(op(1..nos))
9. number of cells	(nla)
10. moisture content of cells	(tf(1..nla))
11. thickness of cells	(tcom(1..nla))
12. slope angle	(slope)
13. zenith angle	(za)
14. surface azimuth angle	(surfacaz)
15. latitude of site	(lat)
16. topographic a/s value	(aos)

CONTINUE

ALGORITHM 20211b : SEARCH SOIL CHARACTERISTICS (BSSS)

```
SEARCH databank facility

    KEY:- soil type

        FOR mean and standard deviation of the following parameters:

            1. saturated hydraulic conductivity
            2. saturated water content
            3. bubbling pressure
            4. percentage sand content
            5. percentage clay content
            6. mineral bulk density

APPLY stochastic variation on parameters 1-6:

            1. lognormal distribution
            2-6. normal distribution

CONTINUE
```

ALGORITHM 2021ic : SOIL - MOISTURE CHARACTERISTIC CURVES (BSSS)

```
DETERMINE
        soil bulk density for each different soil type according to:-

  soil bulk density = (100/(om%/0.224)+((100-om%)/mbd)))

        where   om% = percentage organic matter content
                mbd = mineral bulk density

CONTINUE


DETERMINE
        10 point $\theta$-$\Psi$ curve according to

    FOR each given $\Psi$

        $\theta = a_i + b_i\ (\%\ \text{sand}) + c_i\ (\%\ \text{clay}) + d_i\ (\%\ \text{om}) + e_i\ (\text{sbd})$

        where $\theta$ = soil moisture content
              $\Psi$ = matric suction
              om = organic matter
              sbd = soil bulk density

        coefficients a-e are abstracted from the table in
        section 3.5.3.3 for given values of $\Psi$

CONTINUE
```

```
DETERMINE
        20 point $\theta$-$\psi$ curve with equal $\theta$ intervals    (Howse, 1985)

CONTINUE

CALCULATE

    FOR each soil type, the hydraulic conductivty according to:-

        $k_i = \frac{k_s \, (\theta_i / \theta_j)^c \sum_{j=i}^{m} [(2j+1-21)\psi_j^{-2}]}{\sum_{j=1}^{m} [(2j-1)\psi_j^{-2}]}$

        at equal moisture intervals    (Millington and Quirk).

CONTINUE
```

ALGORITHM 2021 1d : READ METEOROLOGICAL DATA (BSSS)

DEFINE

(a) data input file to be read *interactive system prompted*

READ from input data file the following information:

1. number of hours of rain	(ir)
2. hourly precipitation values	(ppt(1..ir))
3. rain start time	(alr)
4. rain stop time	(amr)
5. barometric pressure (mbs)	(press)
6. albedo of surface	(alb)
7. cloud type	(ncloud)
8. number of air temp observations	(max(1))
9. times of air temp observations	(xxx(1..max(1),1))
10. observations of air temp	(yyy(1..max(1),1))
11. number of rel. humidity obs.	(max(2))
12. times of rel. humidity obs.	(xxx(1..max(2),2))
13. obs. of rel. humidity	(yyy(i..max(2),2))
14. number of cloud cover obs.	(max(3))
15. times of cloud cover obs.	(xxx(1..max(3),3))
16. obs. of cloud cover	(yyy(1..max(3),3))
17. number of ground temp. obs.	(max(4))
18. times of ground temp. obs.	(xxx(1..max(4),4))
19. obs. of ground temps.	(yyy(1..max(4),4))

20. number of wind speed obs.	(max(6))
21. times of wind speed obs.	(xxx(1..max(6),6))
22. obs. of wind speed	(yyy(1..max(6),6))

<u>CONTINUE</u>

ALGORITHM 20211e : ALTERATION OF UNITS (BSSS)

CONVERT

1. air temp observations - centigrade to kelvin,

   i.e. centigrade + 273.15 = degrees kelvin

2. relative humidity (%) to decimal,

   i.e. % * 0.01 = decimal

3. ground temp observations - centigrade to kelvin,

   i.e. centigrade + 273.15 = degrees kelvin

4. wind speed observations - ms^{-1} to $cm\ s^{-1}$,

   $ms^{-1} * 100 = cm\ s^{-1}$

5. slope - degrees to radians,

   i.e. $(degrees * \pi)/180 = radians$

6. surface azimuth - degrees to radians,

   i.e. $(degrees * \pi)/180 = radians$

CONTINUE

CALCULATE

1. rate of increase / decrease throughout day of :-

   a. air temperature

   b. relative humidity

   c. ground temperature

   d. cloud cover

   e. wind speed

2. at each observation time the cumulative amount of :-

   a. air temperature

   b. relative humidity

   c. ground temperature

   d. cloud cove

   e. wind speed

CONTINUE

ALGORITHM 20211f : READ RUN DATA (BSSS)

DEFINE

(a) data input file to be read *interactive system promptec*

READ from input data file the following information:

1. iteration time interval	(af)
2. duration of run time	(immax)
3. julian day	(days)

CONTINUE

ALGORITHM 20211g :NON-ISOTHERMAL EVAPORATION ROUTINE

DETERMINE

evaporation at any time of day and year according to :-

$$\text{evaporation} = \frac{S - I - G - H}{Lx}$$

eq. 20211g.1

where S = incomming radiation to a surface of albedo a_g

I = thermal infra-red radiation

G = heat flux from the ground

Lx = latent heat exchange

H = sensible heat

DETERMINE

S according to:-

$$S = (1-a_g)\,[1-A(u^*,z)](0.349)S_0\cos z$$

eq. 20211g.2

$$+\,(1-a_g)\,[(1-a_0)/(1-a_0\underline{a}_g)](0.651)S_0\cos z$$

where S = incomming radiation at the ground with no cloud cover

a_g = surface albedo

a_g = average ground albedo

a_0 = atmospheric albedo for Raleigh scattering , equal to

$0.085 - 0.247 \log_{10} [(\mathbf{p_s/p_o})\,\mathbf{cos\ z}]$

p_s = surface pressure

p_o = 1000 mb

z = zenith angle of the sun as a function of the time of day and year

A(u*,z) = Mugge - Moller absorption function, equal to $0.271(u^* \sec z)^{0.803}$

u^* = effective water vapour content of the atmosphere

S_0 = solar radiation incident on top of the atmosphere

$(0.349)S_0$ = ammount of solar radiation of wavelength > 0.9 µm

$(0.651)S_0$ = amount of solar radiation of wavelength < 0.9 µm

DETERMINE

u^* according to:- (Smith, 1966)

$$u^* = \exp[0.07074\, T_d + \mathcal{T}]$$ **eq. 2021 1g.3**

where T_d = dew temperature

$\mathcal{T}$ = -0.02290 April - June

$\mathcal{T}$ = 0.02023 all other months

DETERMINE

cloud cover adjustment factor (Haurwitz, 1948)

according to :-

$$CA = (a/94.4).\exp[-m(b-0.059)]$$ **eq. 20211g.4**

where CA = cloud adjustment factor

a&b = empirical coefficients dependent upon cloud type (Balick et al., 1981)

m = secant of the solar zenith angle

DETERMINE

energy reaching the surface (Pochop et al., 1968)

according to :-

$$S_c = S_a - [S_a-(S_a.CA)].CC^2$$ **eq. 20211g.5**

where S_c = energy reaching the surface

S_a = energy reaching surface with no cloud

CA = cloud adjustment factor

CC = visual cloud cover in tenths

DETERMINE

effective incident net insolation according to:-

$$S = S_c.SF$$ eq. 20211g.6

where S = effective incident net insolation

S_c = net insolation

SF = slope factor

DETERMINE

slope factor, SF, according to :-

$$SF = \cos(z).\cos(SI) + \sin(z).\sin(SI).\cos(SAZ-SIAZ)$$

eq. 20211g.7

where SF = slope factor

z = solar zenith angle

SI = slope of the surface

SAZ = solar azimuth angle

SIAZ = azimuth of the slope

DETERMINE

thermal infra-red energy inputs according to:-

(Sellars, 1965)

$$I_{\surd 0} = \mathcal{E}\delta T_a^{4} \left[c + b(e_a^{0.5}) \right]$$

eq. 20211g.8

where $I_{\surd 0}$ = thermal infra-red energy input

E = emissivity (assumed to be 1)

δ = Stephan - Boltzman constant

T_a = shelter air temperature (kelvin)

e_a = water vapor pressure (mb)

b = empirical constant = 0.05

c = empirical constant = 0.61

DETERMINE

water vapor pressure, e_a ,according to:-

(Murray, 1967)

$$e_a = RH. (6.108).exp(A.T_a)/(T_a+273.15-B)$$

eq. 2021 1g.9

where e_a = water vapor pressure

RH = relative humidity (decimal)

T_a = shelter air temperature (kelvin)

A = empirical constant = 17.269

B = empirical constant = 35.86

DETERMINE

infra-red radiation at surface as affected by cloud presence, according to :-

(Sellars, 1965)

$$I_{\surd t} = I_{\surd 0}(1+CIR . CC^2)$$

eq. 2021 1g.10

where $I_{\surd t}$ = cloud adjusted infra-red radiation at surface

$I_{\surd 0}$ = infra-red radiation at surface with no cloud cover

CIR = coefficient dependent upon cloud type (Sellars, 1965 or Oke, 1978)

CC = cloud cover in tenths

DETERMINE

ground radiative emittance, $I^{\wedge}_{t}$, according to :-

$$I^{\wedge}_{t} = E_g \delta (T_g)^4 \qquad \textbf{eq. 20211g.11}$$

where $I^{\wedge}_{t}$ = ground radiative emittance from the surface

E_g = emissivity of the ground

δ = Stephan - Boltzman constant

T_g = ground temperature

DETERMINE

total infra-red input, I, to the surface according to:-

$$I = I_{\surd t} - I_{\wedge t}$$ eq. 20211g.12

where I = total thermal infra-red input to the surface

$I_{\surd t}$ = cloud adjusted infra-red radiation at the surface

$I_{\wedge t}$ = energy radiated from the surface

DETERMINE

the conductive and convective sensible heat transfer , H, according to :- (Lamb, 1974; Oke, 1978)

$$H = -\rho C_p k^2 z^2 \frac{\partial\theta}{\partial z} \frac{\partial v}{\partial z} \text{ SCF}$$ eq. 20211g.13

where H = sensible heat transfer

ρ = air density

C_p = specific heat of dry air at constant pressure

k = von Karman's constant = 0.40

z = observation height

$\partial\theta / \partial z$ = partial derivative of potential temperature w.r.t. height

$\partial v / \partial z$ = partial derivative of windspeed w.r.t. height

SCF is DEFINED by:-

eq. 20211g.14

$SCF = 1.175(1-15R_i)^{0.75}$	when $R_i \leq 0$
$SCF = (1-5R_i)$	when $0 < R_i \leq 0.2$
$SCF = 0$	when $R_i > 0.2$

DETERMINE

potential temperature, O , according to:-

$$\theta = T_a \left(\frac{1000}{P} \right)^{0.286}$$

eq. 20211g.15

where θ = potential temperature

T_a = air temperature

P = air pressure

DETERMINE

the Richardson number, R, according to:-

$$R_i = \left(\frac{g}{\theta} \frac{\partial \theta}{\partial z} \right) / \left(\frac{\partial v}{\partial z} \right)^2$$

eq. 20211g.16

where R = Richardson's number

g = gravity

θ = average potential temperature between the surface and height, z

v = average wind velocity between the surface and height, z.

DETERMINE

latenet heat exchange, Lx, according to:-

$$Lx = -\rho L k^2 z^2 \left(w \frac{\partial q}{\partial z} \right) \left(\frac{\partial v}{\partial z} \right) SCF$$ **eq. 20211g.17**

where Lx = latent heat exchange

ρ = air density

L = latent heat of evaporation

q = specific humidity

v = wind velocity

z = height (i.e. shelter height)

k = von Karman's constant = 0.40

w = saturation factor

SCF = defined in eq. **20211g.14**

ASSUME

heat flux from the ground, G, in temperate latitudes = 0

COMPUTE the following substitutions:-

SUBST. eq. **20211g.3** into eq. **20211g.2** to solve eq. **20211g.2** = S_a

SUBST. eq. **20211g.4** & eq. **20211g.2** into eq. **20211g.5** = S_c

SUBST. eq. **20211g.5** & eq.**20211g.7** into eq **20211g.6** = S

SUBST. eq. **20211g.8** into eq. **20211g.10** = $I_{\sqrt{t}}$

SUBST. eq. **20211g.10** & eq. **20211g.11** into eq. **20211g.12** = I

SUBST. eq. **20211g.15** into eq. **20211g.16** = R_i

SUBST. eq. **20211g.16** into eq. **20211g.14** = SCF

SUBST. eq. **20211g.15** & eq. **20211g.14** into eq. **20211g.13** = H

SUBST. eq. **20211g.15** into eq. **20211g.17** = Lx

SUBSTITUTE equations **20211g.6**
20211g.12
20211g.13
20211g.17 into eq. **20211g.1**

to SOLVE for evaporation.

CONTINUE

ALGORITHM 20211h : ONE DIMENSION INFILTRATION (BSSS)

CALCULATE water movement between cells (**figure 3.3**) according to:-

$$\frac{\partial \Theta}{\partial t} = \frac{\partial}{\partial z}\left(k(\Theta)\frac{\partial \psi}{\partial z}(\Theta)\right) - \frac{\partial k}{\partial z}(\Theta) \qquad \textbf{eq. 20211h.1}$$

SOLVE **eq. 20211h.1** through the following equations.-

Richard's equation for flow

$$q = k(\Theta)\,\Delta h \qquad \textbf{eq. 20211h.2}$$

where q = apparent water velocity

k(Θ) = hydraulic conductivity at soil water content Θ

Δh = hydraulic gradient

Continuity states that:-

$$\frac{\partial \Theta}{\partial t} = \Delta q \qquad \textbf{eq. 20211h.3}$$

where Θ = soil water content

t = time

Δq = flow gradient

Substitute **eq. 20211h.3** into **eq 20211h.2**.-

$$\frac{\partial \Theta}{\partial t} = \Delta\left(k(\Theta)\Delta h\right) \qquad \textbf{eq. 20211h.4}$$

Set

$\Delta h = \psi - z$ **eq. 20211h.5**

where $\psi = \psi(\theta)$ = matric potential at soil water content θ, and may be either a suction or a pressure

z = gravitational head (or depth)

Substitute **eq. 20211h.5** into **eq. 20211h.4**:-

$$\frac{\partial\theta}{\partial t} = \Delta\left(k(\theta)\Delta(\psi - z)\right)$$ **eq. 20211h.6**

Expand **eq. 20211h.6** in Z direction:-

$$\frac{\partial\theta}{\partial t} = \frac{\partial}{\partial Z}\left[k(\theta) \left(\frac{\partial\psi}{\partial z} - \frac{\partial z}{\partial z}\right) \right]$$

$$\frac{\partial\theta}{\partial t} = \frac{\partial}{\partial Z}\left[\frac{k(\theta)\partial\psi(\theta)}{\partial z} - k(\theta) \right]$$

$$\frac{\partial\theta}{\partial t} = \frac{\partial}{\partial z}\left[\frac{k(\theta)\partial\psi(\theta)}{\partial z} \right] - \frac{\partial k(\theta)}{\partial z}$$ **eq 20211h.7**

(One dimensional flow)

CONTINUE

ALGORITHM 202111 :CALCULATE SOIL STRENGTH

DETERMINE

static part of soil strength relationship according to:-

CALCULATE

$m = (4 \times \sin(phi))/(3 \times (1+\sin(phi)))$ **eq. 202111.1**

where phi = internal angle of friction

CALCULATE

$b1 = (\tan(alpha) + \tan(phi)) \times \tan(alpha) \times 8$ **eq. 202111.2**

where alpha = half the cone tip angle = 15^0

CALCULATE using **eq 202111.1**

$a1 = 1 - m$ **eq. 202111.3**

$a2 = 2 - m$ **eq. 202111.4**

$a3 = 3 - m$ **eq. 202111.5**

CALCULATE using **eqs. 202111.3-5**

$b2 = d^2 \times a1 \times a2 \times a3$ **eq. 202111.6**

where d = cone diameter = 0.799 in.

CALCULATE using **eq. 20211i.1**

$b3 = (gg \times (1/\tan(phi)))^{m}$ **eq. 20211i.7**

where gg = shear modulus (set by the user)

CALCULATE using **eq. 20211i.3**

$b4 = (gamma)^{a1}$ **eq. 20211i.8**

where gamma = soil density = 0.067 lb/in^3

CALCULATE using **eqs. 20211i.2, 20211i.5 - 8**

$s1 = (b1 \times b3 \times b4)/b2$ **eq. 20211i.9**

CALCULATE using **eq. 20211i.5**

$d1 = ((c/gamma) \times (1/\tan(phi)) + aa1 + zz)^{a3}$ **eq. 20211i.10**

$d2 = ((c/gamma) \times (1/\tan(phi)) + zz)^{a3}$ **eq. 20211i.11**

CALCULATE using **eq. 20211i.4**

$d3 = ((c/gamma) \times (1/\tan(phi)) + zz)^{a2}$ **eq. 20211i.12**

where c = soil cohesion (user defined)
aa1 = cone length = 1.48in
zz = depth of cone tip (user defined)

CALCULATE using **eqs. 20211i.5 & 20211i.12**

d3 = d3 x a3 x aal **eq. 20211i.13**

CALCULATE using **eqs. 20211i.10, 20211i.11 &20211i.13**

s2 = d1 - d2 - d3 **eq. 20211i.14**

CALCULATE

e1 = c x tan(alpha) x (1/tan(phi)) **eq. 20211i.15**

e2 = ua x tan(phi) **eq. 20211i.16**

e4 = (4 x aal^2 x tan(alpha))/(d^2) **eq.20211i.17**

where ua = air entry matric potential = -17.0 in

CONTINUE

DETERMINE

dynamic section of soil strength relationship
(i.e. the influence of water) according to :-

CALCULATE

e3 = (ua + uw) x tan(phib) **eq. 20211i.18**

where uw = soil water matric potential

phib = remoulded internal angle of
soil friction (user defined)

CALCULATE using **eqs. 20211i.15-18**

s3 = (e1 - e2 + e3) x e4 **eq. 20211i.19**

CALCULATE using eqs. **20211i.9, 20211i.14 & 20211i.19**

ci = (s1 x s2) -s3 **eq. 20211i.20**

CONTINUE

ALGORITHM 20211J : MANAGEMENT OF OUTPUT DATA

```
SET UP
        logical series of output files
        named data.1, data.2, .......data.n,.......data.30
             where 'n' is an integer upto 30

ALLOW
        user to change start 'n' value

MATCH
        output file name with correct input file for multiple runs

WRITE
        RCI values at hourly intervals (or choosen intervals) of top layer for
        duration of run

CONTINUE
```

6.2 DESIGN : Variable source area simulator (VSAS2)

The set of general requirements detailed in **section 1.2** indicate the need to cope with areas that have slope angles greater than say five degrees and are well vegetated (**requirement 1.2e**). The first attempt of the BSSS is unable to simulate the conditions of steep slopes, convergent flow and highly vegetated areas while remaining true to **requirements 1.2a - 1.2d**, i.e. parsimonious data requirements; readily available data; essentially deterministic in nature; and, computationally quick to run (**section 10.2**). In order to fulfil all the requirements as well as possible, it was decided to adapt an existing scheme (VSAS2) to calculate soil strength over steep areas.

The VSAS2 scheme was selected for modification. Although the scheme was originally intended to estimate channel flow from a catchment, the manner in which it simulates the basin through segmentation and the calculation of soil moisture content within each segment on a variable two dimensional grid system (**figure 6.7**), allows a soil strength relationship (**section 4.1**) to be applied to the top layer of a segment.

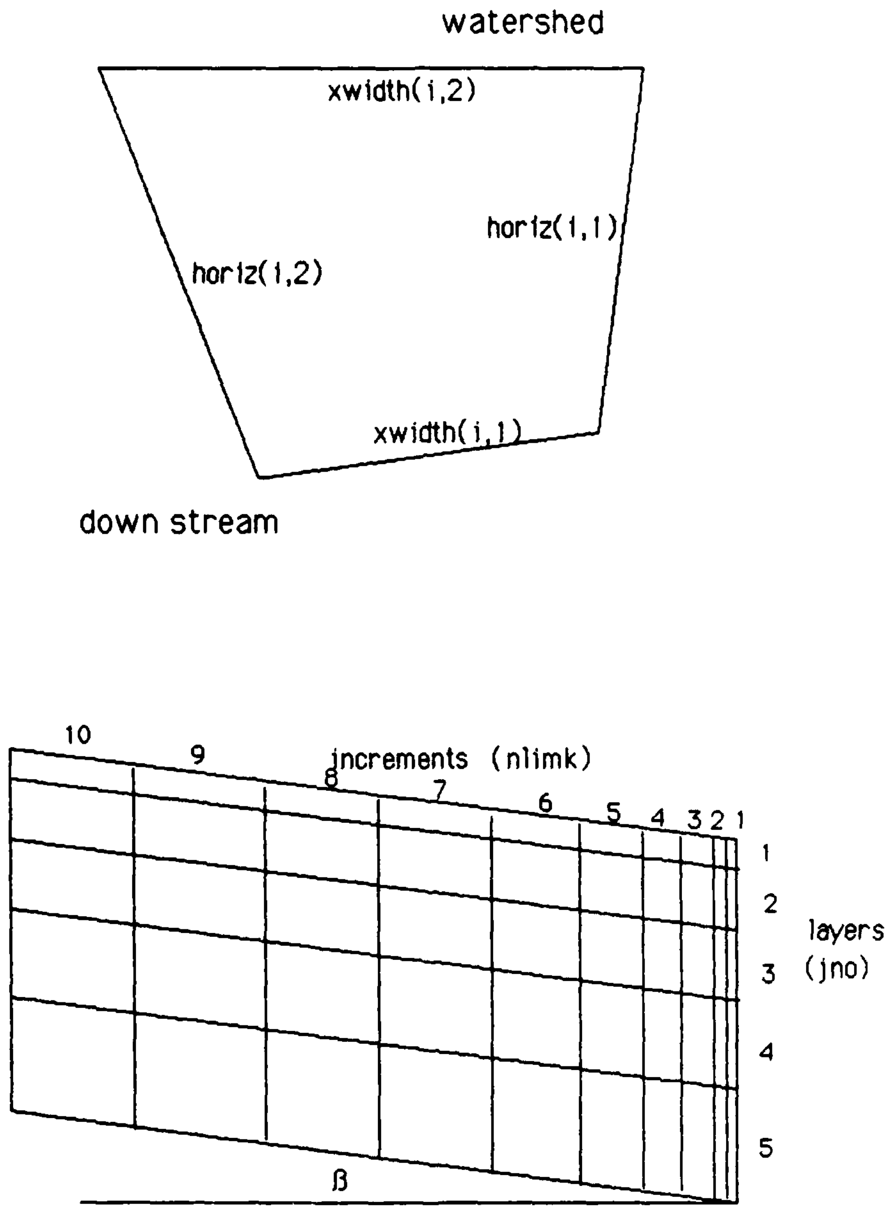
6.2.1 Specification of Detailed Requirements

The following requirements were built into VSAS2 by Bernier (1982):-

6.2.1a The segment / basin is represented as a two dimensional system (**figure 6.7**).

6.2.1b Subsurface flow is allowed to move in non-vertical directions according to computed gradients.

FIGURE 6.7 : PLAN AND SIDE VIEW OF A VSAS2 SEGMENT



6.2.1c The source areas must be capable of both lateral and headward expansion from the established channel system.

The application of VSAS2 for soil strength predictions necessitates a further requirement:-

6.2.1d An appropriate soil strength - soil moisture model should be coupled with the VSAS2 to predict areal RCI values over a segment.

These user requirements form the first stage of the system development shown in **figure 1.2** for the fourth level module 20221 (**figure 1.3**). Using these requirements, a system design can be constructed (**figure 6.8**) which describes a possible solution to the problem. The methodologies of solving the problems are detailed in the program design (**figure 6.9**), which is a further refinement of the system design. The algorithms and data structures to be used are specified in **section 6.2.5**. The program code is then devised according to these algorithms and is further discussed through the program implementation section (**sections 6.2.3** and **6.2.4**).

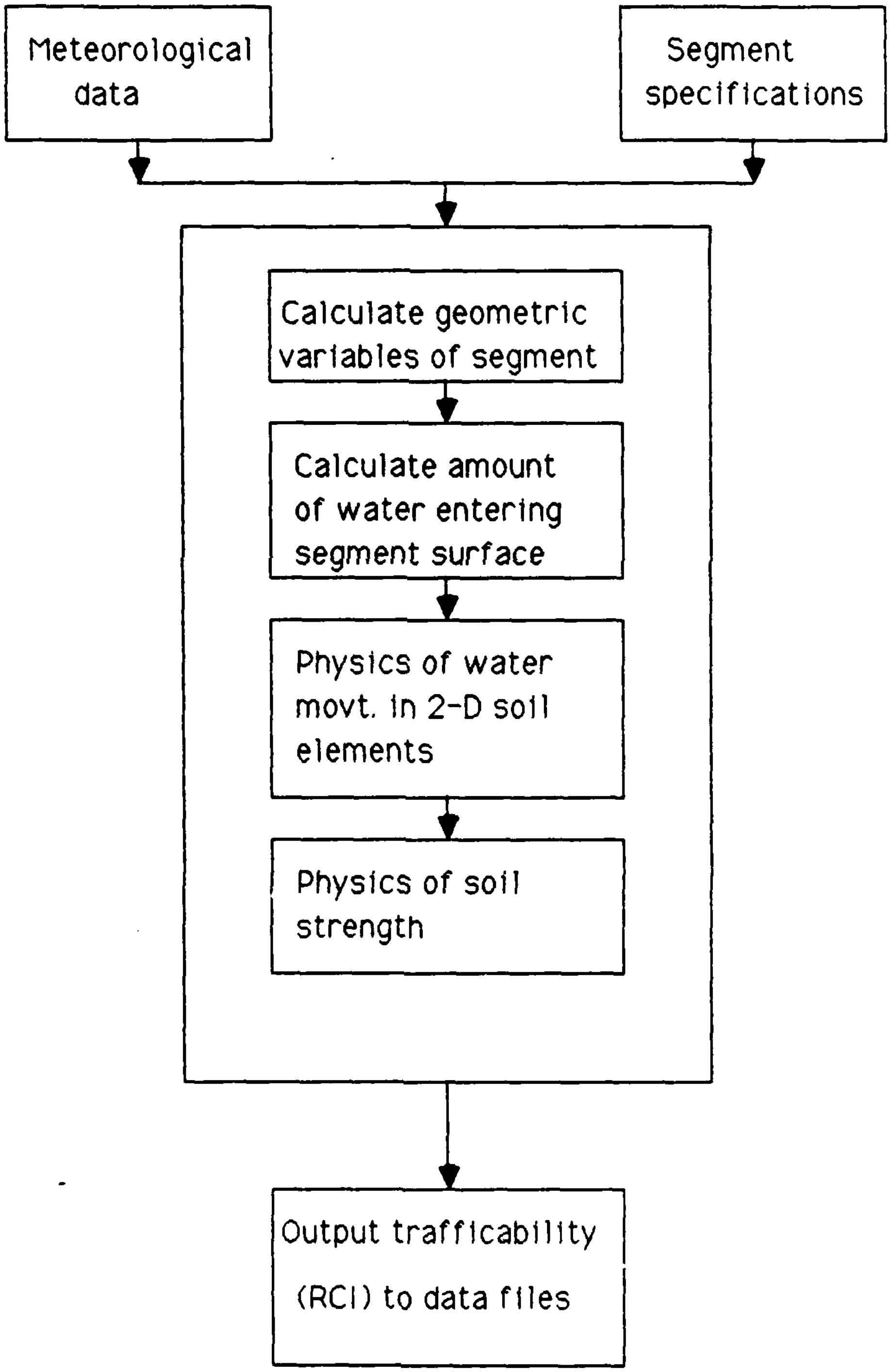
6.2.2 Development of System Design

A general system design for the VSAS2 has been constructed (figure 6.7) from the work carried out by Bernier (1982) and Whitelaw (1987,1988). The design displays modularity with similar modules and sub-modules to the BSSS.

As with the BSSS, the data requirements have been divided into static and

FIGURE 6.8 : VSAS2 - SYSTEM DESIGN

(adapted from Whitelaw, 1988)


Meteorological data
Segment specifications
Calculate geometric variables of segment
Calculate amount of water entering segment surface
Physics of water movt. in 2-D soil elements
Physics of soil strength
Output trafficability (RCI) to data files

dynamic type data, i.e., segment specifications and meteorological data respectively. The information input into the system is then converted into predictions of trafficability averaged over the segment. The 'functional module' transforms the input data by evaluating the amount of rainfall reaching the surface (**algorithm 20221d**), simulating the flow of water through the segment (**section 3.3**) and finally calculating the soil strength according to **section 4.1**. The soil strength for each slope surface element is used to calculate the average soil strength over the segment and is output to the data files. Further refinement of the system design into the program is described in the following section.

6.2.3 Description of Program Design

The modularity of the system design is carried on through to the program design given in **figure 6.9**. The modules have been further refined into smaller sub-modules. Each component of the program design is described by pseudo code (**section 6.2.5**) for the purpose of code construction.

The input requirements have now been split into four:-

(a) segment information describing the physical characteristics of the area to be simulated and described by **algorithm 20221a**,

(b) initial moisture content information for each element of the segment and described in **algorithm 20221e**,

(c) storm information giving hourly rainfall information and described by **algorithm 20221c**, and

(d) soil mechanical properties described in **algorithm 20221j**.

FIGURE 6.9 : VSAS2 - PROGRAM DESIGN

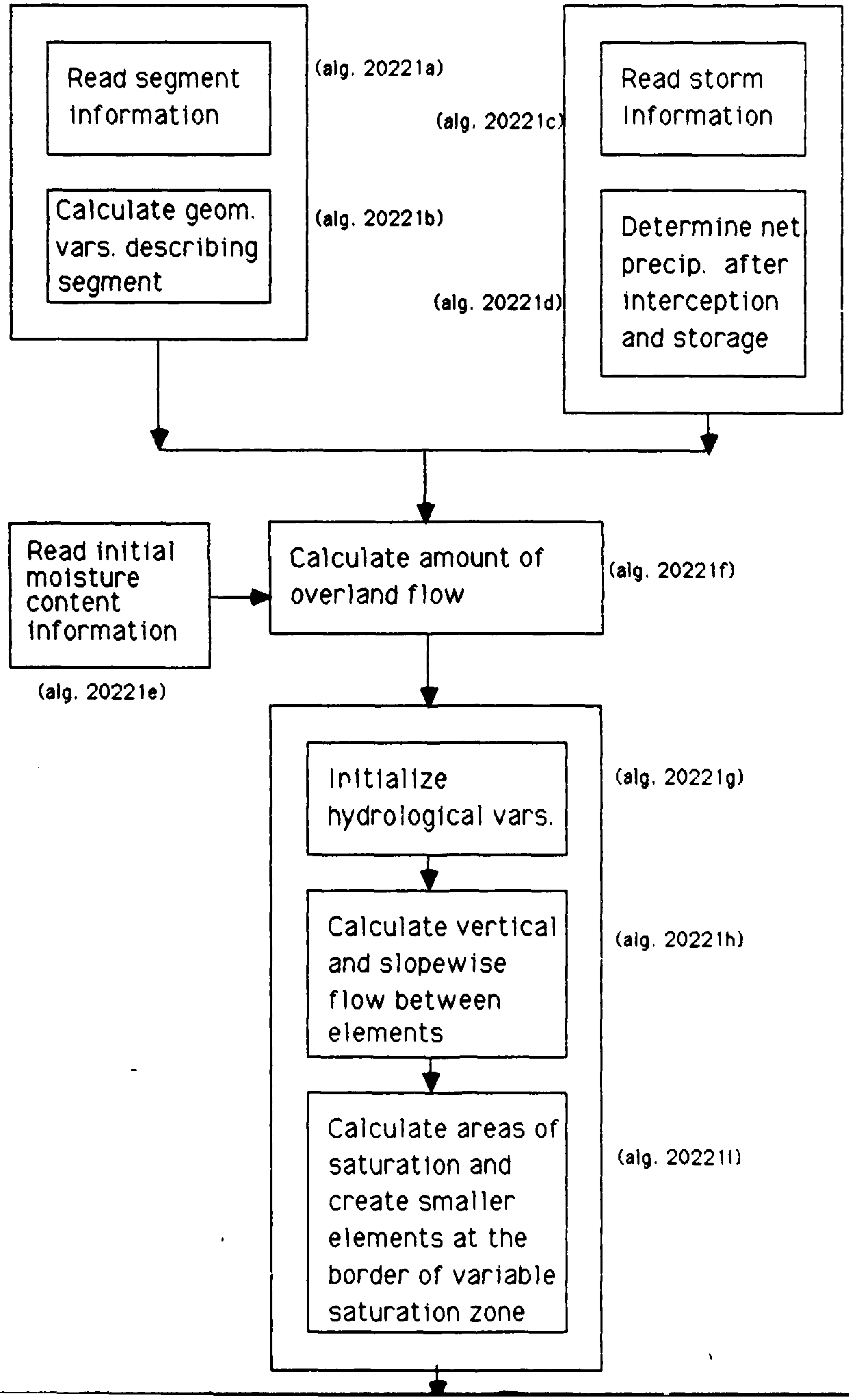
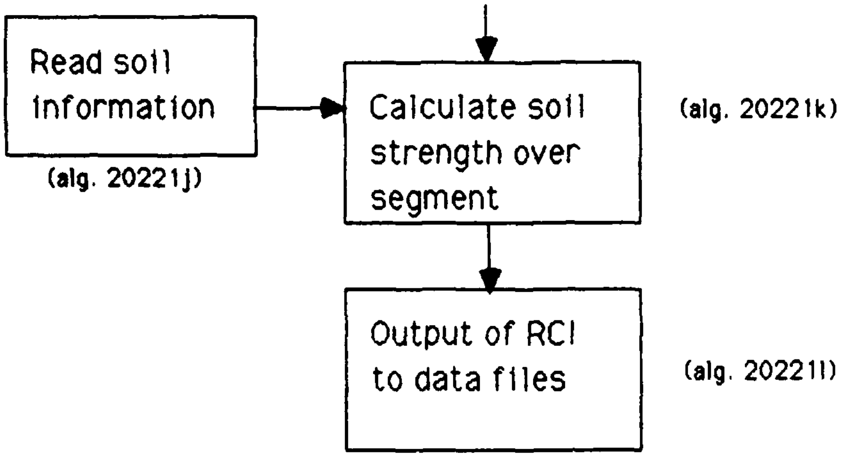

Read segment information
(alg. 20221a)
Calculate geom. vars. describing segment
(alg. 20221b)
(alg. 20221c)
Read storm information
(alg. 20221d)
Determine net precip. after interception and storage
Read initial moisture content information
(alg. 20221e)
Calculate amount of overland flow
(alg. 20221f)
Initialize hydrological vars.
(alg. 20221g)
Calculate vertical and slopewise flow between elements
(alg. 20221h)
Calculate areas of saturation and create smaller elements at the border of variable saturation zone
(alg. 20221i)

FIGURE 6.9 (cont.) : VSAS2 - PROGRAM DESIGN


Read soil information
(alg. 20221j)
Calculate soil strength over segment
(alg. 20221k)
Output of RCI to data files
(alg. 20221l)

The program design in **figure 6.9** shows when this input data is required by the system. The system calculates all the geometric variables describing the segment at the start of the run simulation according to algorithm **20221b**. The amount of water entering the segment is calculated from the rainfall, the amount of water intercepted (**algorithm 20221d**) and the amount of overland flow (**algorithm 20221f**). The VSAS2 scheme makes no allowance for evaporation or evapotranspiration and the limitations of this are further discussed in **section 8.2**.

The following sequence is then carried out at the iteration intervals for the duration of the storm event. After initializing the hydrological variables (**algorithm 20221g**) the simulation of water through the segment is calculated by the two dimensional form of Richard's equation described in **section 3.3**. The grid set-up is altered to create smaller elements at the borders of the saturated area and 75% saturated area (**algorithm 20221i**).

The soil strength is then calculated for the entire segment at the output intervals required by the user according to **algorithm 20221k**. The averaged RCI is then written to the data files (**algorithm 20221l**) for use by the route management scheme (**section 5**).

6.2.4 PROGRAM IMPLEMENTATION :- VSAS2

From the program design (**section 6.2.3**) and algorithm specifications given in **section 6.2.5**, FORTRAN code was written to solve the algorithms. As with the other modules (**figure 1.3**) the algorithms have been coded i; separate subroutines with a controlling subroutine. This section will link the code with the appropriate algorithms and demonstrate how they work together to solve the requirements set out in **sections 1** and **6.2.1**.

The 'subroutine tree' in **figure 6.10** shows the flow of control within VSAS2. The appropriate algorithms coded in the subroutines are given in brackets. All subroutine names used in the VSAS2 begin with a 'v' for ease of identification from the other module subroutines.

As with the VSAS2 obtained from Whitelaw (1987), the flow of the program is controlled through one major subroutine which is called 'vcntl' and can be considered as representing module 2022 in **figure 1.3**. The choice is made here whether to edit a VSAS2 input file in which case 'vmenu' is called, or whether to run VSAS2 when the call to 'vmenu' is by-passed. The subroutine 'vcntl' prompts the user for input files, checks that they exist, asks for an output file for the RCI information and then runs VSAS2 on the information given in the input files. As will be discussed later, VSAS2 provides much more output data than is required for this application to soil strength prediction, but the facility of this extra information will be useful in further development of the scheme

In the application of the VSAS2 within BORTS, it is considered that simulating one segment at a time is the easiest procedure for the user.

Multiple runs of single VSAS2 segments can be used to simulate a larger area if the user requires it.

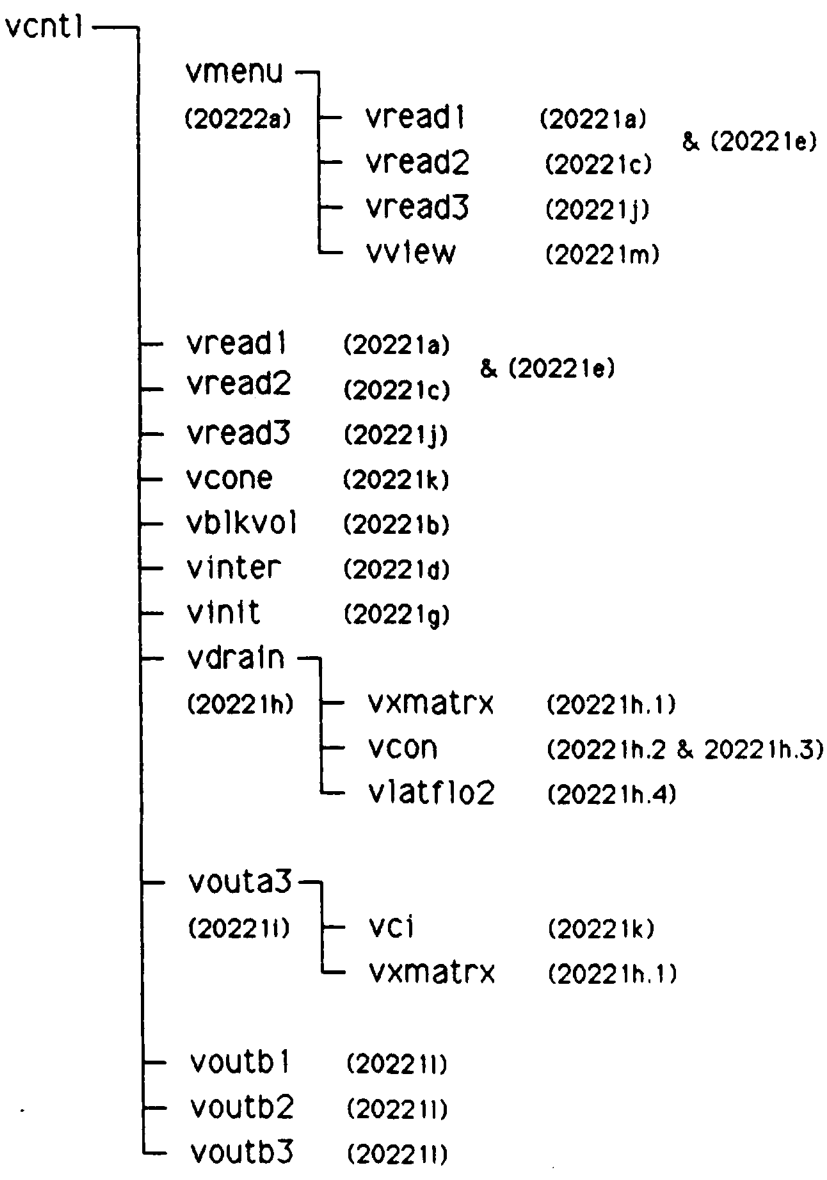
The subroutine 'vread' has three entry point (**figure 6.10**);i.e. 'vread1', 'vread2' and 'vread3', which read the four input files specified by the user. The program then calculates the section of the soil strength relationship (Cochrane, 1987) which is dependent only on the static soil characteristics. The segment geometry (**figure 6.7**) is then calculated by 'vblkvol'. Subroutines 'vcone' and 'vblkvol' are only called once for each segment run.

The subroutine 'vinter' is called next to calculate the net precipitation reaching the surface of the segment. A maximum interception value of 0.249 cm and 0.254 cm for the dormant and growing seasons respectively, is subtracted from the initial rainfall. This storage yields 20% of its volume every hour to the segment as stemflow and drip. A rain free period of twenty-four hours will reset the storage to zero.

The subroutine 'vinit' initializes all the hydrological variables of the soil elements which can vary by layer, depth and position on the slope. Three slope sections (pos) may be specified. The cross-sectional area and the distance for flow between soil element pairs and between the stream side elements and the stream is also calculated for use in 'vdrain' (algorithm 20221h). Initial positive pressures and moisture contents of each soil element is calculated.

The following subroutines simulate the flow of water through the soil over time. Subroutine 'vdrain' calculates the vertical (eq.20221h.4i in algorithm 20221h.4) and slopewise (eq. 20221h.4h in algorithm 20221h.4)

FIGURE 6.10 : VSAS2 SUBROUTINE TREE



flow between a source soil element and a receiving soil element. The receiving element may be a stream. Subroutine 'vdrain' is run on a fifteen minute basis (krep = 4) and the saturated and near saturated elements are handled on a five minute time basis (lrep = 3). The values of 'krep' and 'lrep' are controlled by the user to maintain stability and / or computational time limitations (**section 10.2**).

Within 'vdrain' slopewise flow is calculated first, starting at the lowest layer at the slope base. The sequence of calculation is up the slope and then up the layers. On completion of the slopewise flow calculation, the vertical flow is calculated in the same locative sequence.

Subroutine 'vdrain' calculates the matric potential and hydraulic conductivity of each soil element by calling functions 'vxmatrx' and 'vcon' respectively. Rainfall is added to the surface elements which may temporarily have a water content in excess of their porosity. If this excess water is unable to infiltrate by the end of the 'krep' iteration, extra slopewise flows from the surface layers are calculated. It is assumed that this 'surface water' can travel overland to the stream within the 'krep' iteration - i.e. in 60/krep minutes.

Positive pressures (pres) are calculated according to a 'catching up' procedure where pressures are adjusted to correct flow imbalances that occurred in the final five minutes of the 'krep' iteration.

Subroutine 'vlatflo2' is called by 'vdrain' to calculate the slopewise flow according to eq. 20221h.4h in algorithm 20221h.4. Control is returned to 'vdrain'.

Subroutine 'vdrain' also calls functions 'vxmatrx' and vcon'. Function 'vxmatrx' calculates the matric potential according to Cambell and is detailed in algorithm 20221h.1. Function 'vcon' calculates hydraulic conductivity according to Cambell and is detailed in algorithm 20221h.2. the average hydraulic conductivity between two different soil types is calculated according to Childs and is described in algorithm 20221h.3.

Subroutine 'vouta' has three entry points ;- 'vouta1', 'vouta2' and 'vouta3'. For our specific application, 'vouta3' (algorithm 20221i) is accessed to evaluate the areas of saturation and apply the soil strength subroutine 'vci' to the surface elements. Function 'vxmatrx' is called to calculate the matric suctions used in 'vci'. Subroutine parts 'vouta1' and 'vout2' output the geometrical properties of the segment and are suppressed in this application.

Subroutine 'voutb' also has three entry points and outputs the dynamic quantities calculated in the scheme. Of specific interest in this application, is the output of RCI to a suitable output file. The information output to other files includes the soil water status of each element, flow to a stream on an hourly basis. These output facilities have been retained as they will be useful to further development of the scheme.

To facilitate further development of the VSAS2, **Appendix A** describes the variable, parameter and counter names. Each name is accompanied by a brief description of their use and any other subroutines they are passed to or from.

6.2.5 Specification of algorithms

This section sets out the algorithms used in the VSAS2. The solutions are written in 'pseudo-code', which is an intermediate stage between the description given in English and the program FORTRAN code. This section is represented as stage 3, the program design, shown in **figure 1.2.**

6.2.5a List of algorithms in the following section

Algorithm 20221a :Read segment information.

Algorithm 20221b :Calculate geometric variables describing the segment.

Algorithm 20221c :Read storm information.

Algorithm 20221d :Determine net precipitation after interception, storage and drip.

Algorithm 20221e :Read initial moisture content information.

Algorithm 20221f :Calculate amount of overland flow.

Algorithm 20221g :Initialize hydrological variables.

Algorithm 20221h :Calculate vertical and slopewise flow between elements.

Algorithm 20221i :Calculate areas of 100 & 75 % saturation and create small elements at the border of the variable source area.

Algorithm 20221j :Read soils information.

Algorithm 20221k :Calculate soil strength.

Algorithm 20221l :Output of RCI to data files.

ALGORITHM 20221a :READ SEGMENT INFORMATION (VSAS2)

DEFINE

data file to be read

interactive system prompted

CHECK

data file exists

READ from input data file the following information:

1. number of segments	(mno)
2. title name	(name)
3. a comment note	(note)
4. coarse scale iteration interval	(krep)
5. fine scal iteration interval	(lrep)
6. number of hours between printouts	(npo)
7. option for printout of segment geometry	(nouta)
8. option for calc. of 100 & 75% saturated area	(ntmp)
9. segment number	(isgno)
10. total number of segments	(kno)
11. routing delay in minutes	(irout)
12. number of soil layers	(jno)
13. segment area	(sgarea)
14. channel level	(clevel)
15. stone content of each layer	(stonec(i),i=1,jno)
16. number of increments up the slope	(nlimk(k), k=kno)
17. elevation of each increment	(yi(k,n),k=1,kno n=1,nlimk)
18. depth of each increment	(depmax(k,n), k= 1, kno, n=1,nlimk)

19. distance from stream of each increment	(xi(k,n),k=1,kno n=1,nlimk)
20. width of segment at channel	(xwidth(i,1), i=1, kno)
21. width of segment at watershed	(xwidth(i,2), i=1, kno)
22. length of upstream side of segment	(horiz(i,1), i=1, kno)
23 length of downstream side of segment	(horiz(i,2), i=1, kno)
24. depth of top layer	(adepth)
25. proportion of each layer in remaining depth	(bdepth(i),1=2, jno)
26. proportion of area that is impermeable	(atest)
27. minimum thickness of layer 2	(cdepth)
28. roffst	(roffst)
29. position (%) along slope of zones of changing soil hydrological properties (max of 3)	(pos)
30. mean length of segment	(hlen)
31. cmax of each layer	(cmax1(k,j), k= kno, j=1,jno)
32. porosity of each layer	(poros1(k,j), k= kno, j=1,jno)
33. Cambell coefficient 'a' for each layer	(ca1(k,j), k=1, kno, j=1,jno)
34. Cambell coefficient 'b' for each layer	(cb1(k,j), k=1, kno, j=1,jno)

CONTINUE

ALGORITHM 20221b :CALCULATE GEOMETRIC VARIABLES DESCRIBING SEGMENT (VSAS2)

```
CALCULATE
        longest grid network

CALCULATE
        from longest grid, the division of the       not applicable
        other grids                                  to BORTS
                                                     application
CALCULATE
        angular angular shifts for the non-central
        flow lines for 3-D application

CALCULATE
        the positioning of the soil layers

CALCULATE
        the geometric description of each
        soil element for whole segment

CALCULATE
        total area of simulation segment

CONTINUE
```

ALGORITHM 20221c :READ STORM INFORMATION

```
DEFINE
        data file to be read
                                        interactive system prompted

CHECK
        data file exists

READ from input data file the following information:
        1. date - day                                       (iday)
        2. date - month                                     (imonth)
        3. date - year                                      (iyear)
        4. hourly precipitation values - including all      (precip(i),i=1,
           zero values for entire run period                 run time )

CONTINUE
```

ALGORITHM 20221d : DETERMINE NET PRECIPITATION AFTER INTERCEPTION, STORAGE AND DRIP

```
CALCULATE
          storage capacity of vegetation

WHEN
     rain ≠ 0.0
           CALCULATE
                      if any rain goes into storage
           SET WHEN
                     storage is full all further rainfall goes
                     directly onto catchment

WHEN
     rain = 0.0
           CALCULATE
                      20% of storage is realesed to catchment

WHEN
     24 hour rain free period
           SET
              storage = 0.0

CONTINUE
```

ALGORITHM 20221e :READ INITIAL MOISTURE CONTENT INFORMATION

DEFINE

data file to be read

CHECK

data file exists

READ from input data file the following information:

1. initial moisture content for each element of each layer (pbv(k,n,j), n= 1, nlimk, j=1,jno, k=1,kno)

CONTINUE

ALGORITHM 20221f : CALCULATE AMOUNT OF OVERLAND FLOW

CALCULATE

amount of channel precipitation

CALCULATE

amount of precipitation falling on impervious surfaces

CALCULATE

flow added directly to possible channel in the segment

CONTINUE

ALGORITHM 20221g :INITIALIZE HYDROLOGICAL VARIABLES

```
DETERMINE
            position of all elements

FOR all elements
     SET
          1. coefficients of moisture release curves
          2. coefficients of the unsaturated conductivity curves
          3. porosities
          4. saturated hydraulic conductivities

CALCULATE
            cross-sectional area and distance for flow
            between elements

CALCULATE
            cross-sectional area and distance for flow
            between streamside elements and the stream

CONTINUE
```

ALGORITHM 20221h : CALCULATE VERTICAL AND SLOPEWISE FLOW BETWEEN ELEMENTS

```
CALCULATE
        slopewise flow according to:-
  CALCULATE
          soil moisture content (adding precipitation every hour)
  CALCULATE
          potential (Cambell's method - eq. 20221h.1)
  CALCULATE
          hydraulic conductivity (Cambell's method - eq.20221h.2)
  CALCULATE
          flow parameters for streamside elements
  CALCULATE
          flows (Richards equation - eqs. 20221h.4f & 20221h.4h)
  CALCULATE
          new soil moisture contents of elements
  CALCULATE
          flow into stream

CALCULATE
        vertical flow according to:-
  CALCULATE
          parameters for vertical flow
  CALCULATE
          vertical flow (Richard's equation -eq. 20221h.4i)
  CALCULATE
          new soil moisture contents of elements

CONTINUE
```

CAMBELL'S METHOD OF CALCULATING MATRIC POTENTIAL

$$\psi(\theta) = \psi_e \left(\frac{\theta}{\theta_{sat}}\right)^{-b} \qquad \text{eq. 20221h.1}$$

where $\psi(\theta)$ = matric potential at soil water content θ

θ = soil water content

θ_{sat} = saturated soi water content

ψ_e = air entry potential

b = empirical coefficient

CAMBELL'S METHOD OF CALCULATING HYDRAULIC CONDUCTIVITY

$$k(\theta) = k_{sat} \left(\frac{\theta}{\theta_{sat}}\right)^{2b+2} \qquad \text{eq. 20221h.2}$$

where $k(\theta)$ = hydraulic conductivity at soil water content θ

θ = soil water content

θ_{sat} = saturated soil water content

k_{sat} = saturated hydraulic conductivity

b = empirical coefficient

AVERAGE HYDRAULIC CONDUCTIVITY BETWEEN TWO DIFFERENT SOIL ELEMENTS (CHILDS, 1969)

$$\bar{K} = \frac{D_1 + D_2}{\frac{D_1}{k(\Theta)_1} + \frac{D_2}{k(\Theta)_2}}$$ **eq. 20221h.3**

where $\bar{k}$ = average hydraulic conductivity

$k(\Theta)_1$ = hydraulic conductivity of element 1

$k(\Theta)_2$ = hydraulic conductivity of element 2

D_1 = thickness of element 1

D_2 = thickness of element 2

RICHARD'S EQUATION OF FLOW

$$q = k(\Theta)\,\Delta h$$ **eq. 20221h.4a**

where q = apparent water velocity

$k(\Theta)$ = hydraulic conductivity at soil water content Θ

Δh = hydraulic gradient

CONTINUITY STATES THAT:-

$$\frac{\partial\theta}{\partial t} = \Delta q \qquad \textbf{eq. 20221h.4b}$$

where θ = soil water content

t = time

Δq = flow gradient

Substitute **eq. 20221h.4b** into **eq 20221h.4a**:-

$$\frac{\partial\theta}{\partial t} = \Delta\left(k(\theta)\Delta h\right) \qquad \textbf{eq. 20221h.4c}$$

Set

$$\Delta h = \psi + z \qquad \textbf{eq. 20221h.4d}$$

where $\psi = \psi(\theta)$ = matric potential at soil water content θ, and may be either a suction or a pressure

z = gravitational head

Substitute **eq. 20221h.4d** into **eq. 20221h.4c**:-

$$\frac{\partial\Theta}{\partial t} = \Delta\left(k(\Theta)\Delta(\psi+z)\right)$$

eq. 20221.4e

Expand **eq. 20221h.4e** in x and z directions {* ix = 2 *}

$$\frac{\partial\Theta}{\partial t} = \frac{\partial}{\partial x}\left[\, k(\Theta)_x \left(\frac{\partial\psi}{\partial x} + \frac{\partial z}{\partial x}\right)\right] + \frac{\partial}{\partial z}\left[\, k(\Theta)_z \left(\frac{\partial\psi}{\partial z} + \frac{\partial z}{\partial z}\right)\right]$$

eq. 20221h.4f

From Bernier (1982), x represents slope, i.e. x cos ß

where ß = slope angle

$$x^* = x \cos ß$$

eq. 20221h.4g

Substitute **eq. 20221h.4g** into eq. **20221h.4f**

$$\frac{\partial\Theta}{\partial t} = \cos^2 ß \, \frac{\partial}{\partial x^*}\left[\, k(\Theta)_{x^*} \left(\frac{\partial\psi}{\partial x^*} + \frac{\partial z}{\partial x^*}\right)\right] + \frac{\partial}{\partial z}\left[\, k(\Theta)_z \left(\frac{\partial\psi}{\partial z} + 1\right)\right]$$

eq. 20221h.4h

(Two dimensional flow)

Expand **eq. 20221h.4e** in Z direction:- (* ix = 1 *)

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial Z}\left[k(\theta) \left(\frac{\partial \psi}{\partial z} + \frac{\partial z}{\partial z}\right) \right]$$

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial Z}\left[\frac{k(\theta)\partial \psi(\theta)}{\partial z} + k(\theta) \right]$$

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z}\left[\frac{k(\theta)\partial \psi(\theta)}{\partial z} \right] + \frac{\partial k(\theta)}{\partial z}$$

eq 20221h.4i

(One dimensional flow)

CONTINUE

ALGORITHM 20221i : CALCULATE AREAS OF 100 & 75 % SATURATION AND CREATE SMALL ELEMENTS AT THE BORDER OF THE VARIABLE SOURCE AREA

CALCULATE

elements that are 75 - 100 % saturated

RECALCULATE

grid to give increased spatial resolution at the border of the variable saturation zone

CONTINUE

ALGORITHM 20221j : READ SOILS INFORMATION

DEFINE

data file to be read

interactive system prompted

CHECK

data file exists

READ from input data file the following inform tion:

1. internal angle of friction	(phi)
2. remoulded angle of friction	(phib)
3. soil cohesion	(c)
4. soil shear modulus	(gg)
5. depth of cone tip	(zz)

CONTINUE

ALGORITHM 20221k :CALCULATE SOIL STRENGTH

DETERMINE

static part of soil strength relationship according to:-

CALCULATE

m = (4 x sin(phi))/(3 x (1+sin(phi))) **eq. 20221k.1**

where phi = internal angle of friction

CALCULATE

b1 = (tan(alpha) + tan(phi)) x tan(alpha) x 8 **eq. 20221k.2**

where alpha = half the cone tip angle = 15^0

CALCULATE using **eq 20221k.1**

a1 = 1 - m **eq. 20221k.3**

a2 = 2 - m **eq. 20221k.4**

a3 = 3 - m **eq. 20221k.5**

CALCULATE using **eqs. 20221k.3-5**

b2 = d^2 x a1 x a2 x a3 **eq. 20221k.6**

where d = cone diameter = 0.799 in.

CALCULATE using **eq. 20221k.1**

$b3 = (gg \times (1/\tan(phi)))^{m}$ **eq. 20221k.7**

where gg = shear modulus (set by the user)

CALCULATE using **eq. 20221k.3**

$b4 = (gamma)^{a1}$ **eq. 20221k.8**

where gamma = soil density = 0.067 lb/in^3

CALCULATE using **eqs. 20221k.2, 20221k.5 - 8**

$s1 = (b1 \times b3 \times b4)/b2$ **eq. 20221k.9**

CALCULATE using **eq. 20221k.5**

$d1 = ((c/gamma) \times (1/\tan(phi)) + aa1 + zz)^{a3}$ **eq. 20221k.10**

$d2 = ((c/gamma) \times (1/\tan(phi)) + zz)^{a3}$ **eq. 20221k.11**

CALCULATE using **eq. 20221k.4**

$d3 = ((c/gamma) \times (1/\tan(phi)) + zz)^{a2}$ **eq. 20221k.12**

where c = soil cohesion (user defined)
aa1 = cone length = 1.48in
zz = dept' of cone tip (user defined)

CALCULATE using **eqs. 20221k.5 & 20221k.12**

d3 = d3 x a3 x aa1 **eq. 20221k.13**

CALCULATE using **eqs. 20221k.10, 20221k.11 &20221k.13**

s2 = d1 - d2 - d3 **eq. 20221k.14**

CALCULATE

e1 = c x tan(alpha) x (1/tan(phi)) **eq. 202k1k.15**

e2 = ua x tan(phi) **eq. 20221k.16**

e4 = (4 x aa1^2 x tan(alpha))/(d^2) **eq.20221k.17**

where ua = air entry matric potential = -17.0 in

CONTINUE

DETERMINE

dynamic section of soil strength relationship
(i.e. the influence of water) according to :-

CALCULATE

e3 = (ua + uw) x tan(phib) **eq. 20221k.18**

where uw = soil water matric potential

```
                phib = remoulded internal angle of
                       soil friction (user defined)

    CALCULATE using eqs. 202k1k.15-18

          $s3 = (e1 - e2 + e3) \times e4$                  eq. 20221k.19

    CALCULATE using eqs. 20221k.9, 20221k.14 & 20221k.19

          $vci = (s1 \times s2) - s3$                      eq. 20221k.20

CONTINUE

FOR
    each surface element:-
    CALCULATE
            1. RCI
            2. area

CALCULATE

        average RCI over area according to:-

        $\overline{RCI} = \frac{\Sigma[RCI(element) \times area(element)]}{total\ area}$        eq. 20221k.20

CONTINUE
```

ALGORITHM 202211 : OUTPUT OF RCI TO DATA FILES

```
DEFINE
        name of output file according to the
        logical series of output files named          interactive
        'data.1',.....'data.n', where 'n' may be       system prompted
        any integer up to 30

WRITE
        average RCI value at required output
        intervals for duration of run

WRITE
        calculated grid geometry
                                          suppressed for BORTS
WRITE                                     application
        detailed segment soil water
        status at interval outputs


CONTINUE
```

6.3 DESIGN : Grid Specification

To consider soil strength on an areal basis, a method of representing an area was required. A straightforward way to do this is to impose a grid system over the area (**figure 6.11**). The guidelines in **section 7.5** give more details as how to determine the resolution of the grid and the minimum soil areas to be considered.

6.3.1 Specification of Detailed Requirements

The development of a scheme to define grid specifications was based on the following requirements:-

6.3.1a the grid must be variable in size, i.e. variable number of columns and rows;

6.3.1b the grid should be independent of grid to ground scale, i.e. the user determines the area represented by each square, but this has no effect on the computational running of the scheme;

6.3.1c each grid square is identified by an integer. These integers refer to one run of either the BSSS or the VSAS2, where the output is to a file entitled 'data.n' where 'n' is the identifying integer;

6.3.1d easy editing facilities controlled by the system;

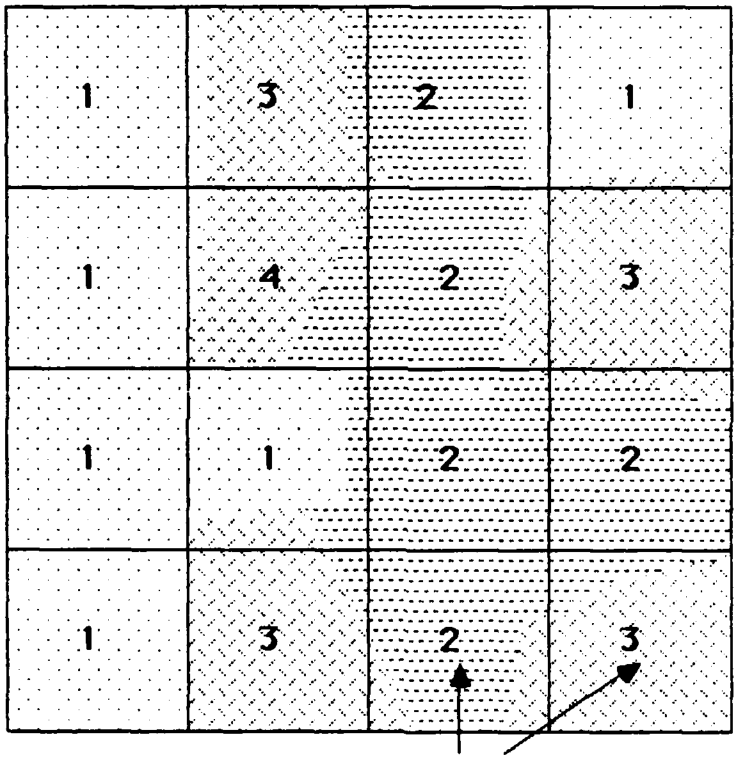
6.3.1e the scheme should be user friendly in terms of:-

(i) system information when choices are to be made,

(ii) system prompts for user response,

(iii) system checking for 'basic' errors, i.e. whether input files exist or if the user really means to overwrite an existing file.

FIGURE 6.11 EXAMPLE OF GRID OVER AN AREA



1	3	2	1
1	4	2	3
1	1	2	2
1	3	2	3

IDENTIFIERS FOR RCI FILES

KEY:

□ Soil Type 1

□ Soil Type 2

□ Soil Type 3

□ Soil Type 4

A solution to these requirements was devised and is shown in **figures 6.12**and **6.14** as the system design. By considering methodologies to implement the solutions of the system design the algorithms were written (**section 6.3.5**) to provide the program design shown in **figures 6.13** and **6.15**. According to these algorithms, program code was constructed (**Appendix B**) and is discussed in the program implementation in **sections 6.3.3** and **6.3.4**.

6.3.2 Development of System Design

General system designs were developed, firstly, for creating a new grid (**figure 6.12**) and secondly, for editing an existing grid (**figure 6.14**). These are not of the same complexity as the schemes described in **sections 6.1, 6.2** and **6.4**, and this is reflected in the small number of modules used to describe the grid specification scheme.

The input requirements are the grid specifications which are then stored in a data file. A new grid is created by the BNGS. The user may then edit any 'identifying integer' within the grid through the EEGS.

6.3.3 Description of the Program Design

The system design of the two schemes is further refined to the program designs shown in **figures 6.13** and **6.15** for the BNGS and EEGS respectively. The two schemes act independently:-

FIGURE 6.12: BNGS - SYSTEM DESIGN

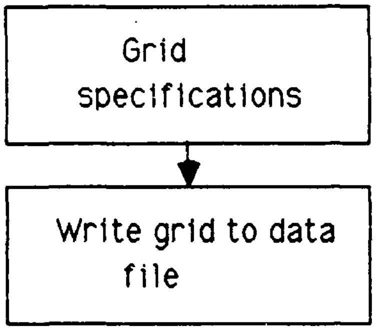


FIGURE 6.13: BNGS - PROGRAM DESIGN

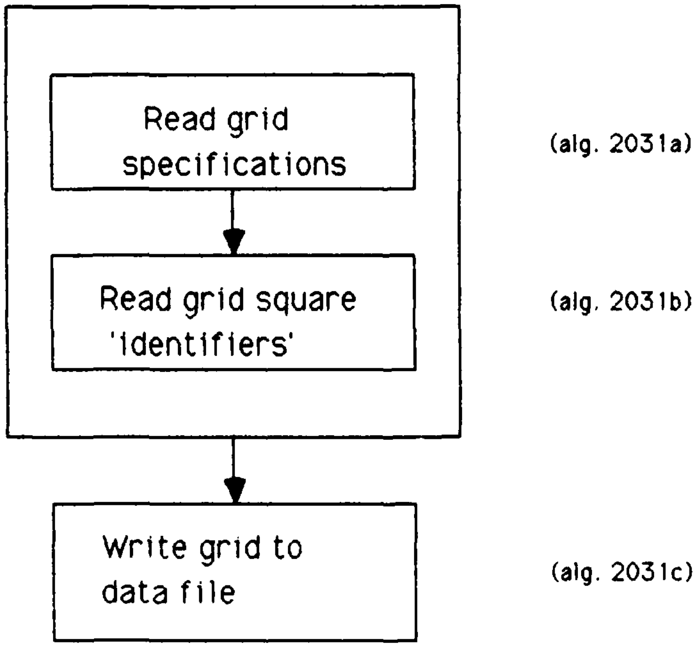


FIGURE 6.14: EEGS - SYSTEM DESIGN

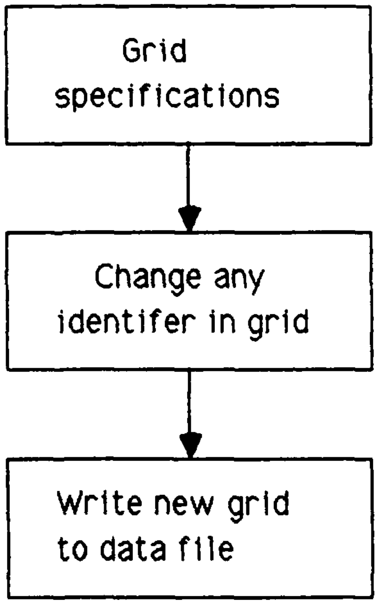
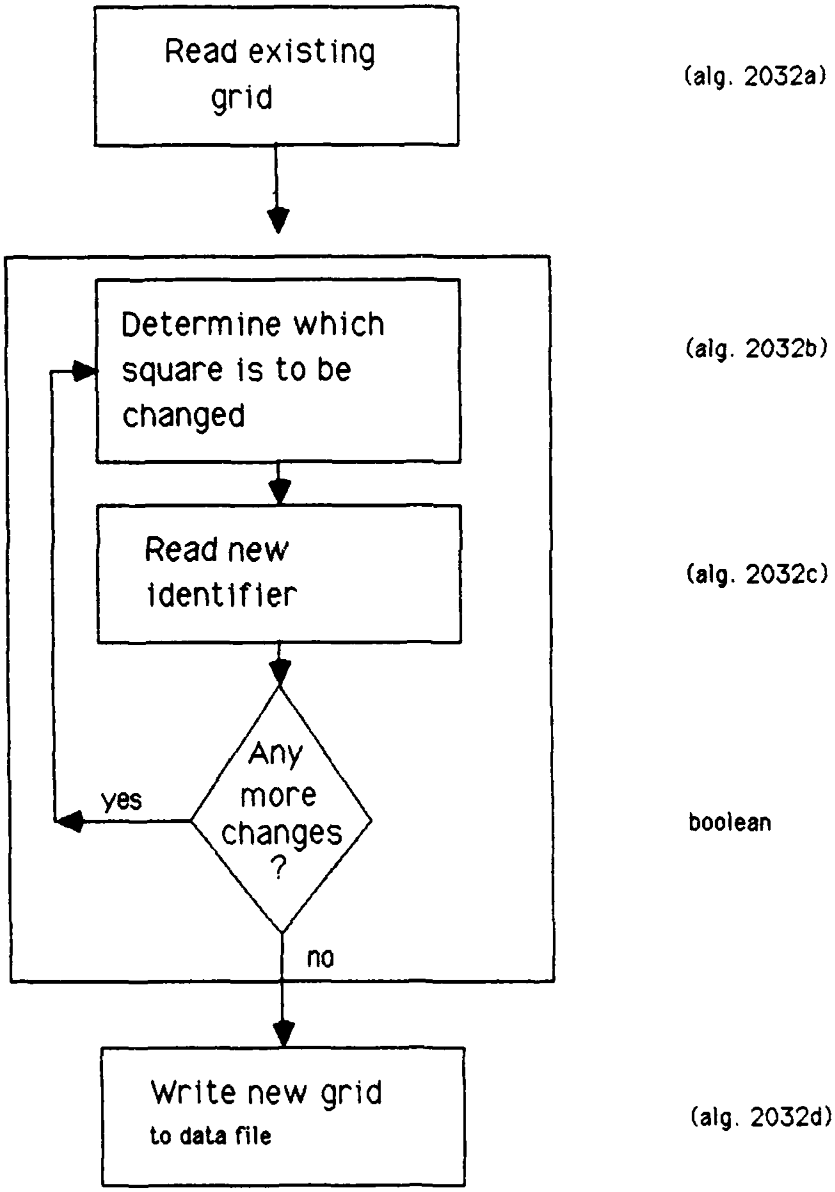


FIGURE 6.15: EEGS - PROGRAM DESIGN



6.3.3a BNGS

The BNGS requires input information regarding the dimensions of the grid (**algorithm 2031a**) and the identifying integer for each grid square (**algorithm 2031b**). The grid specifications are then written to a data file for use by the route management scheme.

6.3.3b EEGS

The EEGS requires an existing grid, that has been created through the BNGS, to be read (**algorithm 2032a**). The system will then ask the user which grid square is to changed (**algorithm 2032b**). For this grid square the new identifying integer supplied by the user will replace the existing one (**algorithm 2032c**). The modified grid is now written to a data file (**algorithm 2032d**). This modified grid can be altered as many times as the user requires.

6.3.4 PROGRAM IMPLEMENTATION :- Grid

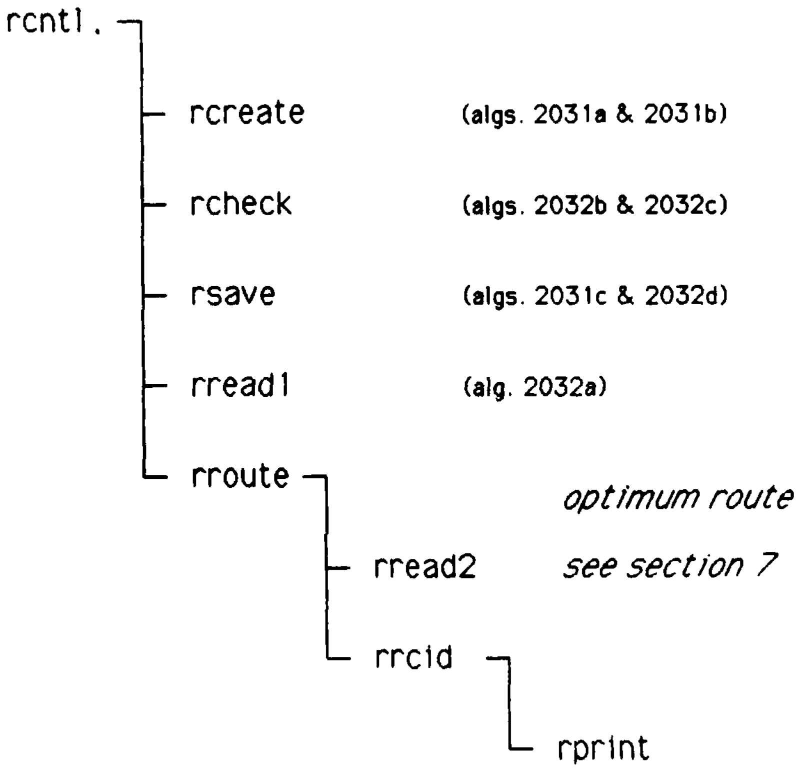
Using the program design and the algorithm specifications detailed in **section 6.3.5**, FORTRAN code was constructed (**Appendix B**) to solve the algorithms. As in the other modules (**figure 1.3**), the consistency in programming style has been maintained through the construction of small separate subroutines called by a controlling subroutine ('rcntl'). This programming style allows the development, replacement or addition of algorithms with ease.

The grid specification scheme has been developed alongside the 'optimum route' management scheme . Although the two schemes are controlled by the same subroutine - namely 'rcntl', they are independent of each other in terms of creating a grid and running the route management scheme. It should be remembered though, that to run the route management scheme a grid and corresponding RCI data files must exist. In this section the subroutines referring to the creation and editing of a grid will be discussed.

From the 'subroutine tree' in **figure 6.16**, it can be seen that subroutine 'rcntl' calls the four subroutines 'rcreate', 'rcheck', 'rsave' and 'rread1'. As their names indicate, 'rcreate' allows the user to create a new grid - each input is prompted by the system. Alterations to an existing grid take place in 'rcheck'. The grid is saved through the subroutine 'rsave' and a grid is read by subroutine 'rread1'.

To facilitate further development of the grid specification scheme, Appendix A describes the variable, parameter and counter names used in the schemes. Each name is accompanied by a brief description of their use and any other subroutines they are passeJ to or from.

FIGURE 6.16: GRID SUBROUTINE TREE



6.3.5 Specification of algorithms

This section sets out the algorithms used in BNGS and EGGS. The solutions are written in 'pseudo-code', which is an intermediate stage between the description given in English and the program FORTRAN code. This section is represented as stage 3, the program design, shown in **figure 1.2**.

6.3.5a List of algorithms used in BNGS

Algorithm 2031a :Read grid dimensions.

Algorithm 2031b :Read 'identifiers'.

Algorithm 2031c :Output of grid.

6.3.5b List of algorithms used in EGGS

Algorithm 2032a :Read data file.

Algorithm 2032b :Edit 'identifiers' of grid squares.

Algorithm 2032c :Read new 'identifier'.

Algorithm 2032d :Output of grid.

ALGORITHM 2031a :READ GRID DIMENSIONS (BNGS)

```
DETERMINE
            1. number of columns   (max. of 30)          system
            2. number of rows      (max. of 30)          prompted

CONTINUE
```

ALGORITHM 2031b :READ 'IDENTIFIERS' (BNGS)

(see **figure 6.11**)

```
FOR
    each grid square:
                    DEFINE                                  system
                          an identifier                     prompted
                    CHECK
                          identifier is an interger

CONTINUE
```

ALGORITHM 2031c :OUTPUT OF GRID (BNGS)

```
WRITE
    grid with 'identifiers' to a data file

CONTINUE
```

ALGORITHM 2032a :READ DATA FILE (EGGS)

```
DETERMINE
            name of grid file to be read                  system
                                                          prompted
READ
      grid into arrays

CONTINUE
```

ALGORITHM 2032b :EDIT 'IDENTIFIERS' OF GRID SQUARES (EGGS)

```
PRINT
      grid and 'identifiers to the screen

DETERMINE
            IF a change in 'identifiers' is required
               THEN DEFINE
                           1. row                         system
                           2. column                      prompted
               ELSE CONTINUE

CONTINUE
```

ALGORITHM 2032c :READ NEW 'IDENTIFIER' (EGGS)

```
DEFINE
        new 'identifier' value                        system
                                                      prompted .
CHECK
        that 'identifier' is an integer

CONTINUE
```

ALGORITHM 2032d :OUTPUT OF GRID (EGGS)

```
DETERMINE
            IF the user wishes to save this grid          system
                THEN DEFINE                               prompted
                        the name for the grid data file
            ELSE CONTINUE

CONTINUE
```

6.4 DESIGN : Route Management

The general requirements set out in **section 1.2** clearly define the need to apply the results of the BSSS and the VSAS2 to trafficability problems. This may be in the form of 'maps' as described by a grid where the information computed by the BSSS and the VSAS2 can be used in route management problems.

6.4.1 Specification of Detailed Requirements

The following requirements were defined as an initial scheme for route management:-

6.4.1a evaluation of RCI over a given route (**figure 6.17a**);

6.4.1b calculation of an optimum route across an area (**figure 6.17b**);

6.4.1c calculation of an optimum route limited by having to go through a particular 'problem' grid square (**figure 6.17c**);

6.4.1d the inclusion of 'time passing' must be inherent in each scheme;

6.4.1e the scheme should be user friendly in terms of:-

(i) system menu information when choices are to be made,

(ii) system prompts for user response,

(iii) system checking for 'basic' errors - i.e. whether input files exist or if the user really means to overwrite an existing file;

6.4.1f data input is derived from the running of the BSSS and /or the VSAS2.

These requirements led to the system design of two types of approach. Firstly, the evaluation of RCI over a given route (**figure 6.18**) and,

FIGURE 6.17 :ROUTE MANAGEMENT SCHEME

COLUMNS

1	2	3	4	
		Start		1
	move 1			2
move 2				3
Finish				4

nodes	1	2	3	4
identifiers	3	1	3	3
times	t1	t2	t3	t4
distances	d1	d2	d3	d4

6.17a What will be the RCI along this route?

Columns

1	2	3	4	
		Start		1
				2
				3
Finish				4

Rows

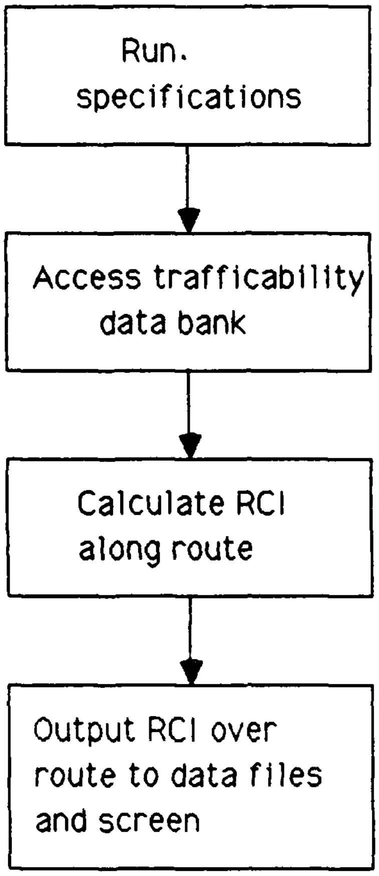
Columns

	1	2	3	4
1			Start	
2			4	
3				
4	Finish			

6.17b Which is the best route?

6.17c Which is the best route that goes through square 4?

FIGURE 6.18: BRES - SYSTEM DESIGN


Run. specifications
Access trafficability data bank
Calculate RCI along route
Output RCI over route to data files and screen

secondly, the 'optimum route' calculation (**figure 6.20**). The 'problem' route being a manipulation of the 'optimum route' calculation.

The two approaches will both access the bank of data files (**figure 6.22**) generated by the running of the BSSS and the VSAS2 at the required time intervals. Output will be in the form of RCI over the specified route for the 'route evaluation scheme', and, in terms of grid square moves for the two variations of the 'optimum route scheme'.

Further refinement of the system designs and devising methodologies to implement the solution are shown in **figures 6.19** and **6.21**. As with the previous schemes, the algorithms are referenced in brackets and are specified in **section 6.4.6.** The code for these algorithms is given in **Appendix B** and the program implementation is discussed in **sections 6.4.3** and **6.4.4**.

6.4.2 Development of System Design

The division of the route management system into two major components necessitates two system designs, namely the 'Bristol Route Evaluation Scheme' (BRES) and the 'Bristol Optimum Route Scheme' (BORS). These are shown in **figures 6.18** and **6.20** respectively.

As with the development of the previous schemes the system designs form a modular framework specifying the flow of information through the system, the general function of the system and the output of the required information. This framework is the developed to formulate the program design discussed in the following section.

FIGURE 6.19: BRES - PROGRAM DESIGN

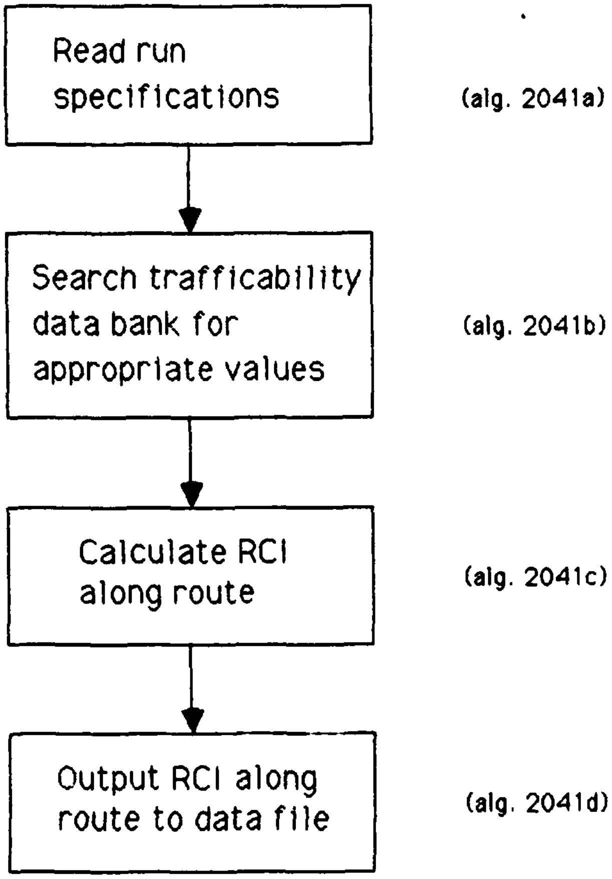

Read run specifications
(alg. 2041a)
Search trafficability data bank for appropriate values
(alg. 2041b)
Calculate RCI along route
(alg. 2041c)
Output RCI along route to data file
(alg. 2041d)

FIGURE 6.20: BORS - SYSTEM DESIGN

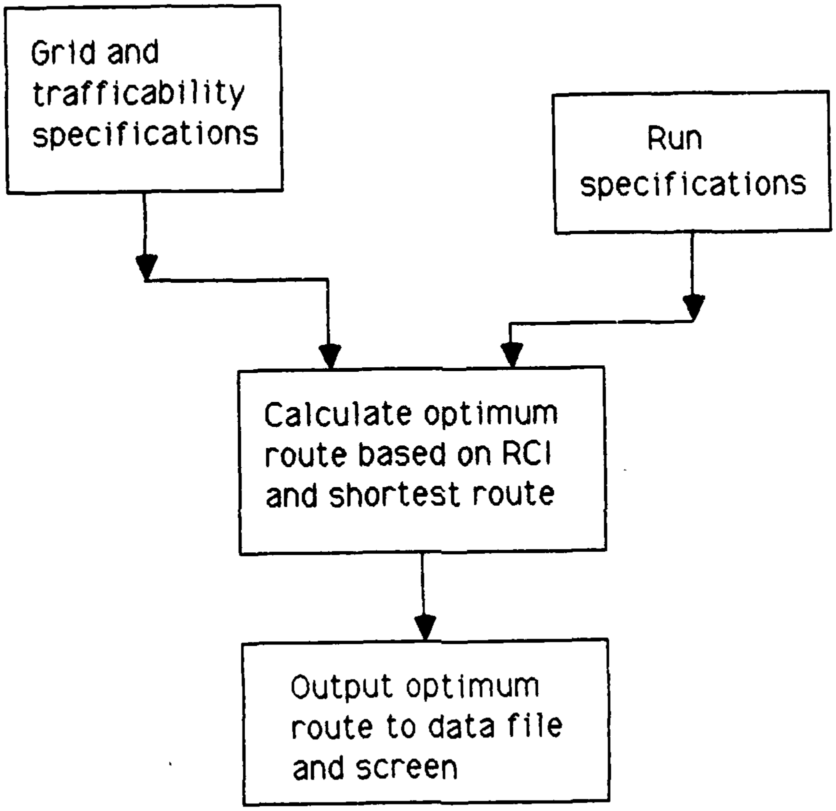

Grid and trafficability specifications
Run specifications
Calculate optimum route based on RCI and shortest route
Output optimum route to data file and screen

FIGURE 6.21: BORS - PROGRAM DESIGN

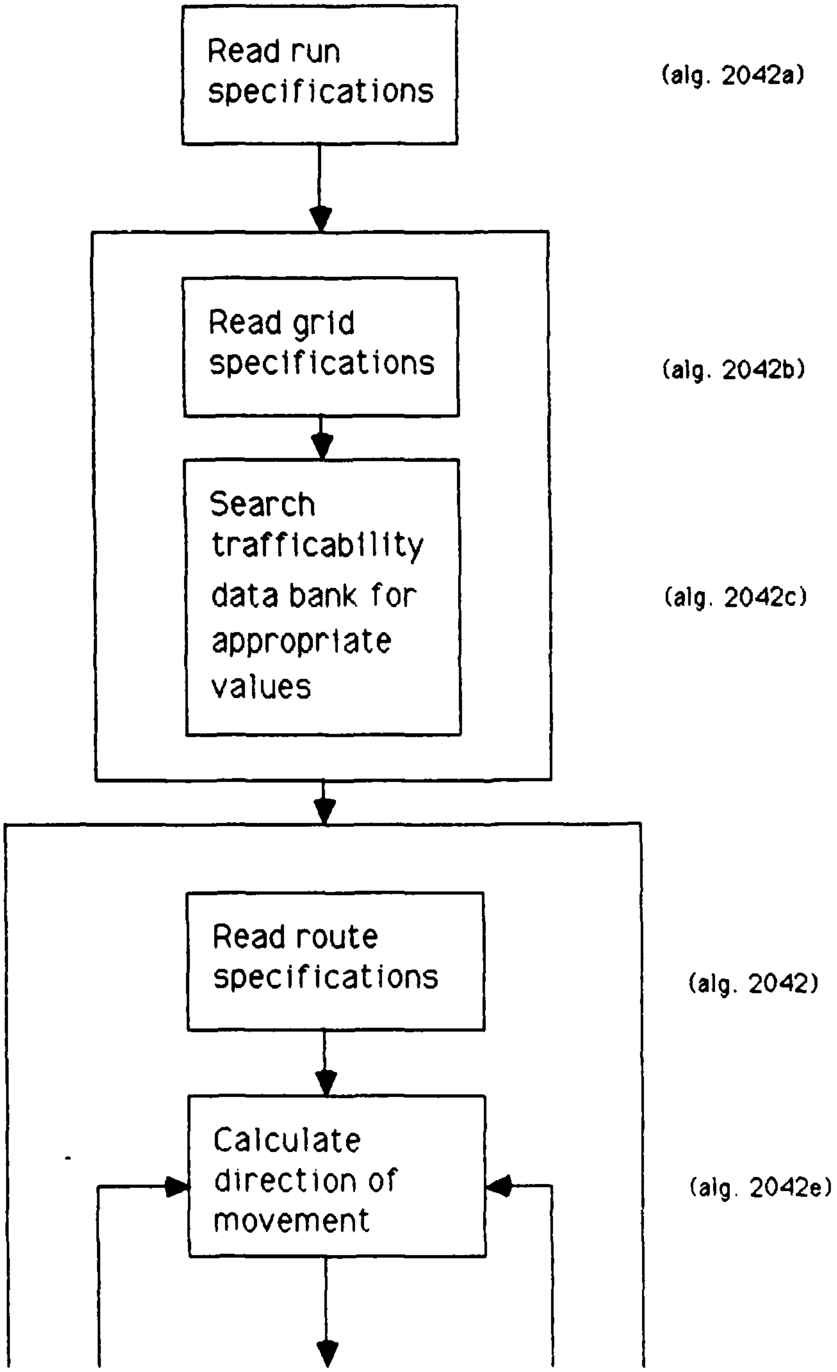

Read run specifications
(alg. 2042a)
Read grid specifications
(alg. 2042b)
Search trafficability data bank for appropriate values
(alg. 2042c)
Read route specifications
(alg. 2042)
Calculate direction of movement
(alg. 2042e)

FIGURE 6.21 (cont): BORS - PROGRAM DESIGN

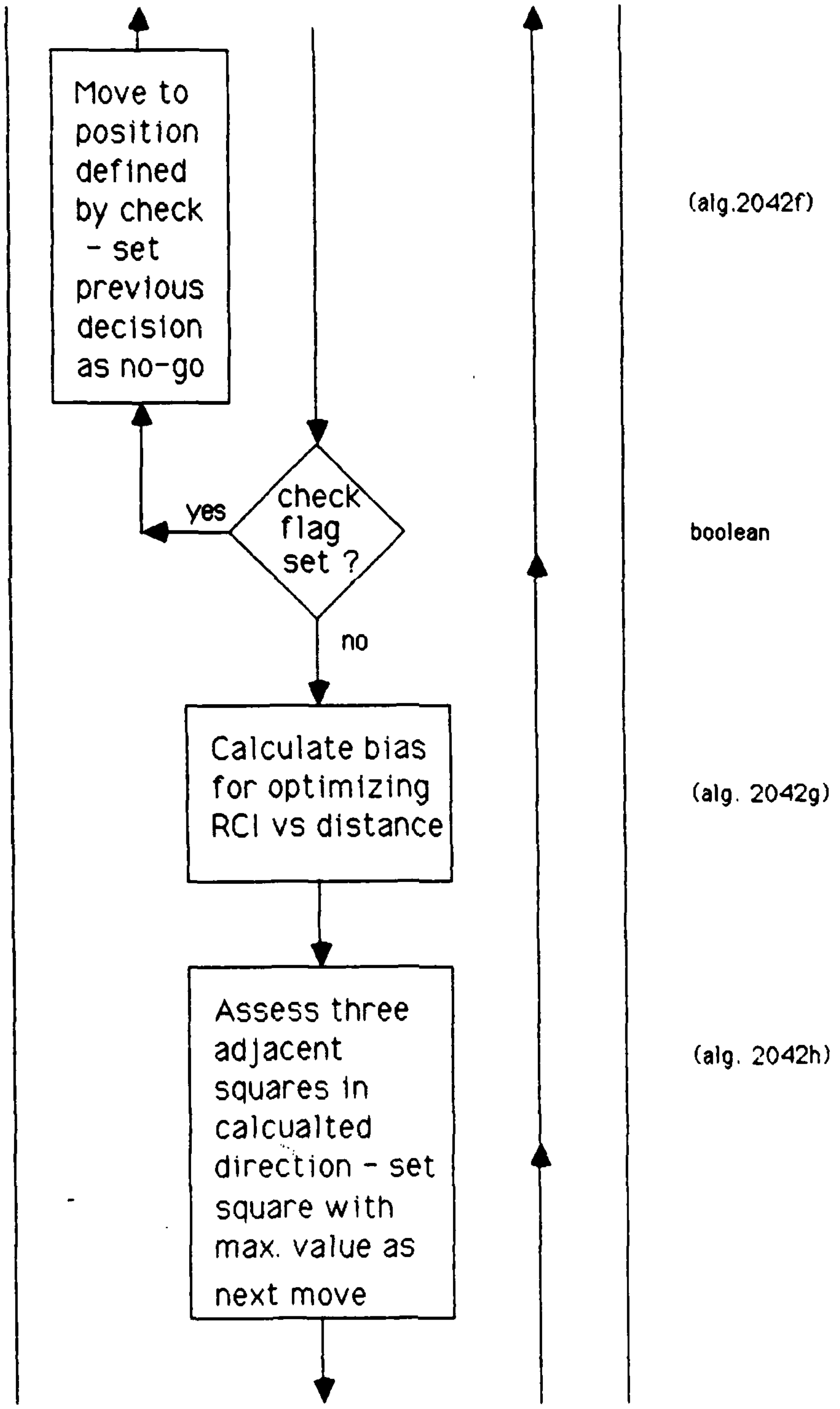

Move to position defined by check - set previous decision as no-go
(alg.2042f)
yes
check flag set ?
boolean
no
Calculate bias for optimizing RCI vs distance
(alg. 2042g)
Assess three adjacent squares in calcualted direction - set square with max. value as next move
(alg. 2042h)

FIGURE 6.21 (cont.): BORS - PROGRAM DESIGN

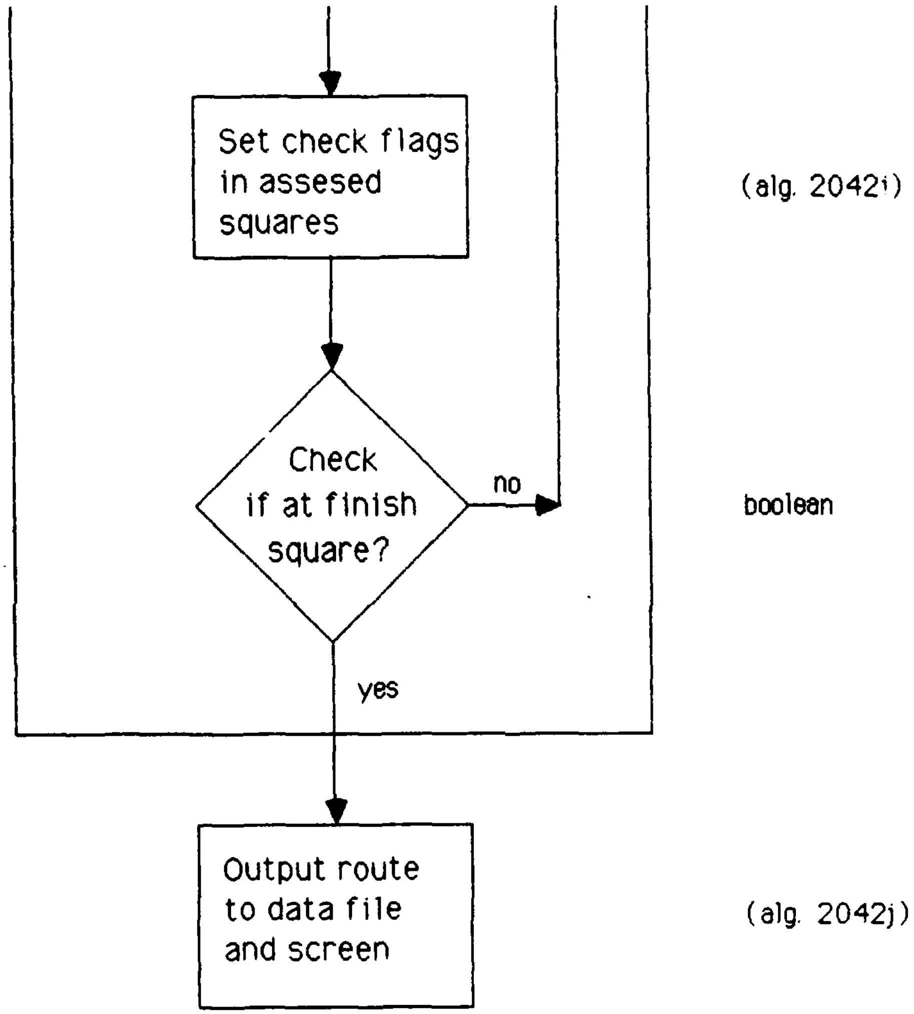
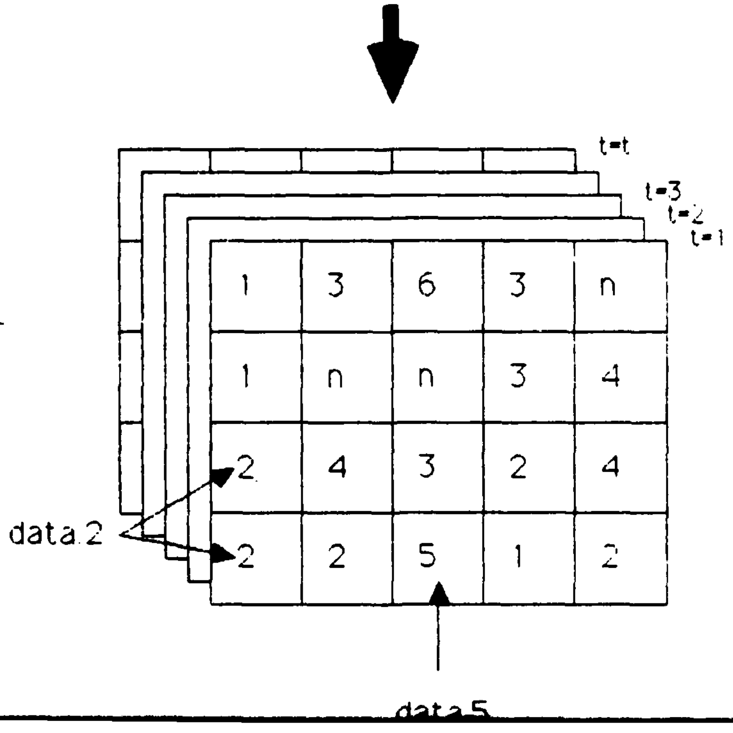

Set check flags in assesed squares
(alg. 2042i)
Check if at finish square?
no
boolean
yes
Output route to data file and screen
(alg. 2042j)

FIGURE 6.22 DATAFILE BANK

	data.1	data.2	data.3	data.4	data.n
time					
1	RCI hour 1	RCI hour 1	RCI hour 1	RCI hour 1	RCI hour 1
2	RCI hour 2	RCI hour 2	RCI hour 2	RCI hour 2	RCI hour 2
	..	..	..	..	..
	..	..	..	..	..
t	RCI hour t	RCI hour t	RCI hour t	RCI hour t	RCI hour t



1	3	6	3	n
1	n	n	3	4
2	4	3	2	4
2	2	5	1	2

6.4.3 Description of Program Design

The system design of the two schemes is further refined to the program designs shown in **figures 6.19** and **6.21** for the BRES and the BORS respectively. The two schemes act independently:-

6.4.3a BRES

The BRES requires information about the route to be evaluated from the user (**algorithm 2041a**). The identifying integer associated with nodes along the route allow the data bank facility (**section 4.4**) to be searched (**algorithm 2041b**) and consequently the RCI along the route to be evaluated (**algorithm 2041c**). The information about the specified route is then written to a data file (**algorithm 2041d**).

6.4.3b BORS

The BORS requires information about starting and finishing positions (**algorithms 2042a** and **2042d**) on a specified existing grid (**algorithm 2042b**). The data bank facility is the searched (**algorithm 2042c**) so that a library of RCI values is built up for the grid (**figure 6.22**). The direction of movement is then calculated according to a combination of highest RCI and shortest distance (**algorithms 2042e - 2042i**). When the finish grid square has been attained the route is written to a data file (**algorithm 2042j**).

6.4.4 PROGRAM IMPLEMENTATION : Route management

From the program design and algorithm specifications given in **sections 6.4.3.** and **6.4.5**, FORTRAN code was constructed to solve the algorithms. As with the previous modules (**figure 1.3**) the consistency in programming style has been maintained by the construction of small separate subroutines called by a controlling subroutine. In this case the route management scheme has been divided into two, that is, the 'evaluation of RCI over a route' has been developed separately (**figure 6.18**) and the 'optimum route' calculation has been developed along w'th the grid specification scheme (**figure 6.20**). This occurred because the 'optimum route' calculation depends on the grid specifications, whereas, the 'evaluation of RCI' over a route depends on the user defining the route precisely (**figure 6.17a**) in terms of nodes, distance, time and 'identifier' of each node alor ; the route.

6.4.4a Evaluation of RCI over a route (BRES)

The 'subroutine tree' shown in **figure 6.23** demonstrates the flow of control with the appropriate algorithms coded in the subroutines given in brackets. All subroutines within this scheme begin with a 't' and can be considered as representing module 2041 in **figure 1.3**. Subroutine 'tcntl' accesses only subroutine 'tred' which has two entry points - namely 'tred1' and 'tred2', which create a new route and read an existing route respectively. The small amount of data file searching and the output of the results is carried out in 'tcntl' itself.

FIGURE 6.23: ROUTE EVALUATION SUBROUTINE TREE

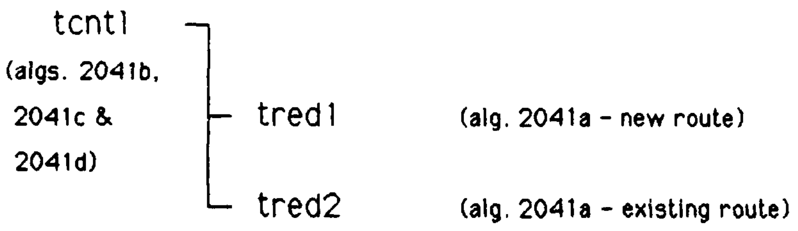
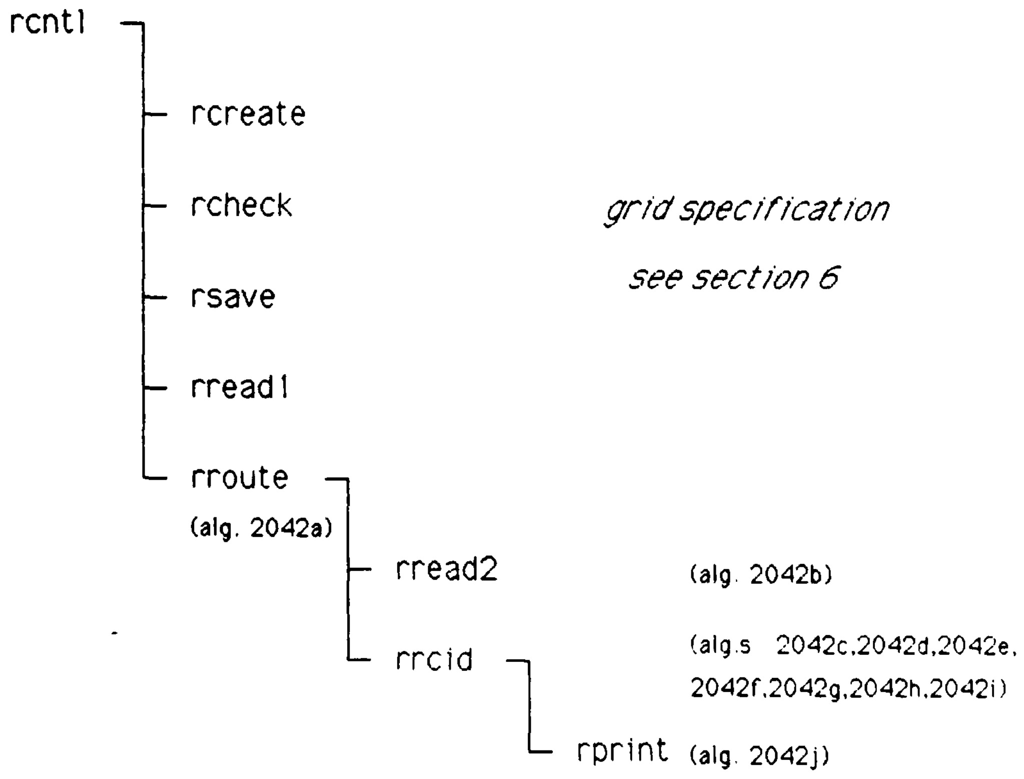


FIGURE 6.24: OPTIMUM ROUTE SUBROUTINE TREE



6.4.4b Optimum route (BORS)

The 'subroutine tree' shown in **figure 6.24** indicates the flow of control, with the appropriate algorithms coded in the subroutines given in brackets. As discussed in the 'grid specification' section, subroutines 'rcreate', 'rcheck', 'rsave' and 'rread1' refer to algorithms defining the grid. Subroutine 'rroute' , as its name implies, controls the 'optimum route' scheme. Subroutine 'rroute' calls two subroutines, namely 'rread2' to read the grid specifications, and, 'rrcid' which searches the bank of data files created by the BSSS and the VSAS2. The optimum route is calculated according to algorithms 2042c -2042i. Subroutine 'rrcid' calls subroutine 'rprint1' to output the route in terms of a series of grid square moves.

To facilitate further development of the route management scheme, **Appendix A** describes the variable, parameter and counter names. Each name is accompanied by a brief description of their use and any other subroutines they are passed to or from.

6.4.5 Specification of algorithms

This section sets out the algorithms used in BRES and BORS. The solutions are written in 'pseudo-code', which is an intermediate stage between the description in English and the program FORTRAN code. This section is represented as stage 3, the program design, shown in **figure 1.2**.

6.4.5a List of algorithms used in BRES

Algorithm 2041a :Read run specifications.

Algorithm 2041b :Search data bank.

Algorithm 2041c :Calculate RCI along route.

Algorithm 2041d :Output of evaluated route.

6.4.5b List of algorithms used in BORS

Algorithm 2042a :Read run specifications.

Algorithm 2042b :Read grid specifications.

Algorithm 2042c :Search data bank.

Algorithm 2042d :Read route specifications.

Algorithm 2042e :Calculate direction of movement.

Algorithm 2042f :Reset to eliminate loops in route.

Algorithm 2042g :Calculate bias.

Algorithm 2042h :Assess three grid squares for next move.

Algorithm 2042i :Set check flags.

Algorithm 2042j :Output of route information.

ALGORITHM 2042a :READ RUN SPECIFICATIONS (BORS)

```
INFORM
        the user that a databank of trafficability
        files is expected to exist

ALLOW
       the user to return to previous level's menu

DETERMINE
          1. time of run                              system
          2. maximum number of files                  prompted
             (data.1,...data.n) to be accessed

CONTINUE
```

ALGORITHM 2042b :READ GRID SPECIFICATIONS (BORS)

```
DETERMINE
           name of file to be read                    system
                                                      prompted
READ
       grid file into arrays

CONTINUE
```

ALGORITHM 2042 c :SEARCH DATA BANK (BORS)

```
FOR
    each grid square
        SEARCH
              data bank with
                  KEYS:- idenifier and time

STORE
      vaiues in grid format

CONTINUE
```

ALGORITHM 2042d :READ ROUTE SPECIFICATIONS (BORS)

```
DETERMINE
          1. start square                          system
          2. finsih square                         prompted
              DEFINED by
                           (a) row
                           (b) column

CONTINUE
```

ALGORITHM 2042e :CALCULATE DIRECTION OF MOVEMENT (BORS)

```
FROM
     the current position to the finish square
        CALCULATE
                      the dominant direction of movement
                      i.e. north <-> south
                       or west <-> east

CONTINUE
```

ALGORITHM 2042f :RESET TO ELIMINATE LOOPS IN ROUTE (BORS)

```
EVALUATE
           'check flag' - this refers to the step when one of the
            assessed squares was previously considered
    IF
     'check flag' is set
           RETURN
           -       to grid square defined by 'check flag'
           SET
               previous decision to 'no-go' and re-assess

CONTINUE
```

ALGORITHM 2042g :CALCULATE BIAS (BORS)

```
CALCULATE
          1. column bias :-  biasc = finish col - present col
          2. row bias    :-  biasr = finish row - present row
          3. total bias  :-  total = biasc +biasr
          4. diff. bias  :-  max(biasc,biasr) + 1

DETERMINE
          the adjustor to be applied to each square
          considered according to :-

          (a) north <-> south
             squares:
             1. same col, row + 1       adjustor = 1
             2. col + 1, row + 1        adjustor = (biasr - 1) / total
             3. col - 1, row + 1        adjustor = (biasr - 1) / total

          (b) west <-> east
             squares:
             1. col + 1, same row       adjustor = biasc / total
             2. same col, row + 1       adjustor = biasr / total
             3. col + 1, row + 1        adjustor = diff / total

ADJUST
       each RCI value of the three squares under assessment

CONTINUE
```

ALGORITHM 2042h :ASSESS THREE GRID SQUARES FOR NEXT MOVE (BORS)

```
COMPARE
        the three adjusted squares

SET
    square with maximum value as next step in route

CONTINUE
```

ALGORITHM 2042i :SET CHECK FLAGS (BORS)

```
FOR
    the three assessed squares
        SET
            their 'check flag' equal to the step identification value

CONTINUE
```

ALGORITHM 2042j :OUTPUT OF ROUTE INFORMATION (BORS)

```
WRITE
      steps of calculated route to
          1. data file
          2. screen

CONTINUE
```

ALGORITHM 2041a :READ RUN SPECIFICATIONS (BRES)

```
CHOICE
        of:- 1. create new route
             2. read existing route

ON (1)
      DEFINE new route in terms of:
           (a) number of nodes on route
           (b) distance of nodes along route                 system
           (c) 'identifier' of each node                     prompted
           (d) time step at each node
           (e) name of file to save route on

      SAVE
            route on named file

CONTINUE

ON (2)
        DEFINE
           (a) number of nodes in route                   system
           (b) name of data file that the route           prompted
           -   is to be retrieved from

CONTINUE

CONTINUE
```

ALGORITHM 2041b :SEARCH DATA BANK (BRES)

```
FOR
    each node:
              SEARCH
                     data bank with
                          KEYS:- identifier and time

STORE
     RCI values in arrays

CONTINUE
```

ALGORITHM 2041c :CALCULATE RCI ALONG ROUTE (BRES)

```
STORE
      in arrays values FOR
                          1. node number
                          2. distance along route
                          3. RCI value

CONTINUE
```

ALGORITHM 2041d :OUTPUT OF EVALUATED ROUTE (BRES)

```
WRITE
     arrays 1-3  in algorithm 2041c to an output file
```

6.5 Discussion and Conclusions

The benefits of adopting a clearly defined development strategy have been discussed in **section 1.4**. This section, through the adoption of that strategy, has documented stages 1 -4 in **figure 1.2** for the design of the components of the BORTS. Each component has had the requirements stated in **section 1.2** further refined, a system design formulated on the basis of the detailed requirements, a program design developed, computer code constructed and subsequently discussed in the program implementation.

The extent of the modules or components developed in this section are:-

(a) The BSSS (**section 6.1**), which is based on an existing infiltration and evaporation scheme. The errors in the original scheme, such as the omission of correct albedo handling, and coding errors were corrected. New development, as discussed in **section 4**, was incorporated into the scheme.

(b) The VSAS2 (**section 6.2**) was adapted to calculate soil strength over the segment. Surperfluous output was restricted though is still available for further development. The scheme was made more user friendly according to **section 4.4**. Requirements for the scheme (Bernier, 1982) have been combined with the requirements of this project and as inadequate documentation for the VSAS2 scheme existed, system and program designs have been drawn up in this section. The VSAS2 code (Whitelaw,1987) has remained as unaltered as possible and the manner in which it solves the two dimensional infiltration equation (**section 3.3**), hydraulic conductivity, matric potential etc. remains the same. It is considered that having a basically unaltered VSAS2 scheme as an option in the BORTS will be of great benefit for initial comparisons (**sections 10.2** and **10.3**). Further development of the VSAS2 scheme will be an important aspect to the second generation BORTS model outlined in **section 10.4**.

(c) The creation of a grid (**section 6.3**) for the route management scheme has been developed specifically for application to this project.

(d) The development of two route management schemes (**section 6.4**) for specific application to this project. These included a route evaluation scheme and an optimum route scheme which could also be applied to problem routes as discussed in **section 5**.

The use of this section is primarily for further development, tracing errors as discussed in **section 1.4.8** and general maintenance of the system. The progression through the development of each component is one that becomes increasingly more complicated allowing scrutiny at different levels of complexity. This flexibility for the developer or maintainer is an important aspect to software and system quality.

7 GUIDELINES FOR THE USER

7.1 Input data for the BSSS

The input data requirements for the BSSS have been split into three sections:-

7.1.1 Meteorological Data (algorithm 20211a)

All these variables, with the exception of albedo, are readily available, measurable data. The input of albedo for different types of surfaces may be estimated from **figure 7.1**. Cloud type is clarified in **figure 7.2**.

7.1.2 Site Information (algorithm 20211d)

The description of an example soil profile is given in **figure 6.2**. The evaluation of the topographic index, aos, is derived from calculating a/s, where a = area drained per unit contour length, and, s = local slope angle. Areas that are described as flat with little area draining, i.e. low a/s values are allocated an 'aos' value of 1.0. Areas that are described as slopes with a certain amount of area draining, i.e. medium a/s values are allocated an 'aos' value of 1.2. Areas that are described as flat with a larger area draining into it, i.e. high a/s values are allocated an 'aos' value of 0.8. These rules are summarized in **section 7.5**. Surface azimuth is abstracted from the topographic map.

FIGURE 7.1 : RANGE OF ALBEDO FOR A VARIETY OF SURFACES

SURFACE	ALBEDO RANGE
Short grass	0.30-0.24
Kale	0.27-0.19
Oak (foliage)	0.30-0.16
Oak (bare)	0.25-0.13
Soils	0.05-0.40
Crops	0.18-0.25
Clearing	0.195-0.135

FIGURE 7.2 : CLOUD TYPE KEY

INDEX VALUE	CLOUD TYPE
1	Cirrus
2	Cirrostratus
3	Altocumulus
4	Altostratus
5	Stratocumulus
6	Stratus
7	Nimbostratus
8	Fog

FIGURE 7.3 SOIL TYPE KEY

INDEX	SOIL TYPE
1	Silt
2	Silty loam
3	Loam
4	Sandy loam
5	Loamy sand
6	Silty clay loam
7	Clay loam
8	Sandy clay loam
9	Silty clay
10	Sandy clay
11	Clay

7.1.3 Operational Data (algorithm 20211f)

Julian day and duration of run time are straightforward for the user to define but the setting of the iteration time step for the inexperienced user may be problematic. Obviously the smaller the iteration time step, the less likelihood there is of mathematical instability. On the other hand, the larger the iteration time step, the quicker the scheme will be to run. It is recommended that the iteration time step is chosen from the range 10s - 120s. The frequency of output requirements will also influence the choice of the time step, e.g. output every fifteen minutes would indicate that a lower iteration time step might be more appropriate.

7.2 Output Data for the BSSS

For the application of the BSSS to route management problems, the output from the BSSS of hourly RCI values must be stored in a data file entitled 'data.n', where 'n' is an integer from 1 to 30, and refers to an identification procedure used in both the grid and route schemes. There is extra output readily available for the developer to check the scheme or adapt the scheme for other applications.

7.3 Input Data for the VSAS2

The input data requirements for the VSAS2, can be divided into four categories:-

7.3.1 Soil Strength Characteristics (algorithm 20221j)

These are the same as required by the BSSS, and are easily measurable.

7.3.2 Initial Moisture Contents (algorithm 20221e)

Initial moisture contents for each element (**figure 6.7**) must be provided by the user.

7.3.3 Storm Characteristics (algorithm 20221c)

The storm characteristics for each hour of the run duration must be input, even if zero. The program will accept lines of blanks, where each blank line represents eight hours of no rain.

7.3.4 Segment Specifications (algorithm 20221a)

Figures and show the geometric description of the segment required to run the VSAS2. The other information required but not shown in figures and includes stone content, level of the top layer above the channel, and area of impervious surface area. These are all described in **figure 6.7** and **section 6**. Also included in this input section are the coefficients and saturated hydraulic conductivity for use in calculating the soil moisture characteristic curves. The evaluation of these input is through extra information and / or user experience.

7.4 Output Data for the VSAS2

The output of RCI information is sent to files entitled 'data.n', where 'n' is the identifying integer. The RCI is an average of the RCI over the entire segment. There is also extra output data available - describing the geometry of the segment and detailed soil water information throughout the segment at hourly intervals. This has been restricted for this application but is easily available to aid further development and applications.

7.5 Setting Up the Grid For Use In Running the BSSS, VSAS2 and Route Management Scheme

1. Identify major soil types (1..11, **figure 7.3**).
2. Identify any 'no-go' areas.
3. Identify major 'a/s' areas, where a = area drained per unit contour length, and s = local slope angle. (n.b. number of 'a/s' categories = AS)
4. Identify major soil area as % of 'go' area (MAX).
5. Solve equation 8.5.1 with the results from 3 and 4 to determine the minimum soil area (MIN) to be considered.
6. Identify all soil areas with an area greater than MIN.
7. Calculate maximum possible number of runs =

   (number of soil areas x AS)
8. Compare value calculated in 7 with computer restricted run number - adjust MIN until number of runs is within the acceptable restricted run number.
9. Identify each grid square with an identifying 'run number (**figure 6.11**)

10. Set up on the computer:

    (a) input files for all runs (menu driven),

    (b) grid description (menu driven).

11. Run package (menu driven).

Equation 7.5.1: First approximation of minimum soil area to consider.

$$MIN = \frac{AS(100-MAX)}{(R-AS)}$$

where MIN = % minimum soil area to be considered

MAX = % maximum area of major soil type

AS = number of 'a/s' categories

R = maximum number of computer runs available

7.6 Evaluate RCI Over a Route

Figure 6.17a shows an example route to be evaluated. To evaluate this route the computer will ask the user to define the following information:-

1. number of nodes along the route,

2. for each node:-

   (a) distance along the route,

   (b) time, t, of the output in 'data.n' to be used,

   (c) identifying integer, 'n', for access to file 'data.n'.

7.7 Optimum Route Management

This scheme expects s full set of data files to be available for every 'identifyer' specified in the chosen grid (**figure 6.11**). The system will prompt the user for start and finish grid squares. The scheme outputs the calculated route as a series of grid steps.

8 Assumptions and Limitations of the BORTS

8.1 BSSS

The development of the BSSS according to the requirements set out in **sections 3.2, 3.4, 4** and **6.1** necessitated making some assumptions. These assumptions are necessary because of the complex nature of the real world processes which the submodels used in the BSSS are trying to simulate.

8.1.1 Infiltration Scheme

The one-dimensional infiltration scheme adopted for use in the BSSS is described in **section 3.2**. An important aspect of adopting any scheme is to specify the situations where it can and cannot be used. The circumstances in which Darcy's law can be applied are as follows:-

8.1.1a The body being modelled must be large in comparison to the microstructure, i.e. approximately a metre or more deep is adequate but when assessing the flow of water to a plant root from the volume of soil associated with it which may only be a few cm s, the application of Darcy's law would be inappropriate (Hillel, 1980).

8.1.1b The velocity of flow must be small. This is assessed by the evaluation of Reynold's number:

$$R = vr\rho/\eta \qquad \textbf{eq. 8.1}$$

where R = Reynold's number

v = mean velocity of flow

r = mean pore size

ρ = density of the fluid

η = viscosity of fluid

When the Reynold's number exceeds 1000 the flow is no longer laminar and as the Stokes-Navier equation on which Darcy's law is based is only applicable to laminar flow it must also be true that Darcy's law will no longer be applicable. It was found by Fancher, Lewis and Barnes (1933) that the restriction of Darcy's law should be to when Reynold's number is less than 1. In the majority of applications Reynold's number takes a value in the order of 0.1.

8.1.1c It ignores the internal structure of the conducting medium, i.e., it is modelled as a uniform medium with the flow uniformly distributed over the cross-section.

8.1.1d No account is taken of the hystersis effect encountered in the wetting and dryying of soils.

8.1.1e It is assumed that Richard's equation accurately describes vertical movement of water in the soil profile. The effect of macropores, piping or crusting is not considered.

8.1.1f The effect of a wide range of temperatures is not considered.

8.1.1g When the rain intensity exceeds the infiltration capacity of the soil and the soil surface detention is exceeded, it is assumed that the extra rainfall runs directly off into a river system and is not available for infiltration on its passage to the river. With the inherent variability of both hydrological characteristics of any given soil type and the soil moisture content across a surface area, this may be an important restriction especially in the two-dimensional application.

8.1.2 Evaporation Scheme

The calculation of evaporation by the non-isothermal routine described in **section 3.4**. The main restriction of the routine is in evaluating the albedo parameters used. **Section 4.2** sets out the problem in more detail and provides a methodology for helping the non-expert evaluate the albedo of surfaces. The scheme assumes that the albedo does not alter in value as the soil changes moisture content.

Although the scheme calculates evaporation it does not account for evaporation from vegetation or transpiration from vegetation. There is also no evaluation of interception by the vegetation canopy. The implication of these restrictions is that the scheme is really only applicable to areas where transpiration is considered small in comparison to thé evaporation. In effect this limits application to temperate or colder type climates.

8.1.3 Soil Characteristics Databank

The development of the soil characteristic data bank has a major inherent restriction in that a soil is described as one of the eleven USDA soil types. Given that restriction, the soil characteristic parameters are calculated from distributions described by Rawls and Brakensiek (1982) and from their results. The values used to describe each soil type do not include the effects of organic matter content or changes in porosity.

8.1.4 Soil Strength Scheme

The calculation of CI is well defined in **section 4.1**, but the calculation of RCI as defined by the U.S. Army (1959) has been assumed to be represented by:-

$$\text{RCI} = \text{CI} \times \text{Remoulded Index}$$

where the remoulded index is assumed to be equal to 0.7.

This assumption is based on a 'typical' value across the range of USDA soil types. In reality the remoulding index has been found to have the following ranges (U.S. Army, 1959):-

(a) Inorganic clays of high plasticity	0.75 - 1.35
(b) Clayey gravels, gravel-sand-clay mixtures, clayey sands, inorganic clays of low to medium plasticity, silty clays	0.45 - 0.75
(c) Silty gravels, silty sands, inorganic and organic clays and silts, fine sands, clayey silts	0.25 - 0.85

8.2 VSAS2

The restrictions of the infiltration scheme used in the BSSS also apply to the two-dimensional scheme used in the VSAS2. As the same method of calculating soil strength the restriction of the RCI evaluation also applies. The other assumptions and limitations of the VSAS2 are as follows:-

8.2.1 Evaporation

There is no calculation of evaporation or evapotranspiration within the VSAS2 scheme though there is an empirically based system for modelling the interception of the vegetation canopy and treating the canopy as a temporary storage of the intercepted water which is released at a set rate dependent on the size of the store.

8.2.2 Spatial Variability

The VSAS2 can only simulate the spatial variability of meteorological or soil variables through the input data files. This entails much work for the user.

8.3 Route Management

The route management scheme through its use of a grid assumes that an area can be adequately modeled by the resolution of the grid. This is alleviated through the development of a hierarchy of grid systems which concentrate on either problem areas or areas of special interest.

The use of the RCI library requires careful 'housekeeping' by the user and can be considered as a restriction to the inexperienced user. The accessing of the RCI library is conditional upon distance travelled and the speed of the vehicle. In this first generation route management scheme an average speed across a route is asked for. From the work carried out from the DRIVEB model an improvement of the scheme is outlined in section 10 which will include the automatic calculation of speed from RCI and therefore the system will calculate when the next set of RCI files should be accessed.

8.4 Conclusions

The BORTS is a first attempt to develop a scheme with the requirements set out in **section 1.2**. As such, it was important to develop a strategy (**section 1.4**) which would allow the main processes to be modelled in a manner that could be considered as independent. This independence allows any of the schemes to be easily replaced or upgraded by improve methods.

The complex nature of the real world has led to the development of models that simulate physical processes in a simplified manner. This has necessitated making assumptions and imposing limitations on the use of the models. The development of the BORTS has used several well established procedures and has developed several new methods of solving physical relationships. Although none of these procedures are perfect as is shown by the assumptions and limitations listed above, they do attempt to deal with the main physical processes. The independent nature of the development strategy will enable several of these restrictions to be lifted after further development.

It should of course be noted that although one of these restrictions might be solved, the new set of restrictions accompaning a new procedure may be greater. There is also a very real possibility that the increased sophistication of a new methodology may negate some of the requirements of the overall system. When compared to when and where the system can be used with confidence, the above restrictions are not considered excessive.

9 : HANDS ON TEACHING FACILITY

The following section assumes that all the subroutines have been compiled and a master control file has been created (see **appendix C** for details of how to compile the package). In this case the package has been called 'master'. The data files required throughout this teaching facility either must exist or you are guided through the steps to create new data files.

It is not necessary for the user to work through this teaching facility in one go - the user can end a session anytime the master menu appears on the screen and the option '5 FINISH' is selected.

The teaching facility will demonstrate the following procedures:-

9.1 Run one BSSS file.

9.2 Run three BSSS files.

9.3 Run one VSAS2 file.

9.4 Apply results from 9.1 - 9.3 to:

(a) evaluate a given route,

(b) calculate optimum route.

9.5 Edit a BSSS input file and run.

9.6 Edit a VSAS2 input file and run.

9.7 Edit a grid to incorporate new files.

9.8 Apply results 9.1 -9.3 & 9.5 - 9.7 to:

(a) evaluate a given route,

(b) calculate optimum route.

9.1 Run One BSSS File

```
master
 ************************************************
 WELCOME TO THE BRISTOL ROUTE PACKAGE
 ************************************************
 ENTER ANY NUMBER TO CONTINUE
2
 MASTER MENU
   1. ACCESS BRISTOL RCI PROGRAM
   2. ACCESS VSAS2 RCI PROGRAM
   3. OPTIMUM ROUTE PROGRAM
   4. EVALUATION OF RCI OVER A GIVEN ROUTE
   5. FINISH
 INPUT OPTION 1-5
1
 MENU SELECTION
 1. CREATE A NEW DATA FILE
 2. CHANGE AN EXISTING DATA FILE
 3. RUN BRISTOL SSS
 4. EXIT WITHOUT A RUN
3
 WHICH DATA FILE IS TO RUN?
y10.data
 OUTPUT FOR THIS RUN WILL BE IN data.    1
 ALL OUTPUT FILES ARE data.n, WHERE n MAY BE CHANGED
 DO YOU WISH TO CHANGE n? y or n
n
```

```
DO YOU WANT TO MAKE ANOTHER RUN? y or n
n
 MASTER MENU
   1. ACCESS BRISTOL RCI PROGRAM
   2. ACCESS VSAS2 RCI PROGRAM
   3. OPTIMUM ROUTE PROGRAM
   4. EVALUATION OF RCI OVER A GIVEN ROUTE
   5. FINISH
 INPUT OPTION 1-5
1
```

9.2 Run Three BSSS Files

```
MASTER MENU
   1. ACCESS BRISTOL RCI PROGRAM
   2. ACCESS VSAS2 RCI PROGRAM
   3  OPTIMUM ROUTE PROGRAM
   4. EVALUATION OF RCI OVER A GIVEN ROUTE
   5. FINISH
 INPUT OPTION 1-5
1
 MENU SELECTION
 1. CREATE A NEW DATA FILE
 2. CHANGE AN EXISTING DATA FILE
 3. RUN BRISTOL SSS
 4. EXIT WITHOUT A RUN
3
 WHICH DATA FILE IS TO RUN?
y10.data
 OUTPUT FOR THIS RUN WILL BE IN data.  1
 ALL OUTPUT FILES ARE data.n, WHERE n MAY BE CHANGED
 DO YOU WISH TO CHANGE n? y or n
y
 INPUT VALUE FOR n 1-30
2
 DO YOU WANT TO MAKE ANOTHER RUN? y or n
y
 WHICH DATA FILE IS TO RUN?
wrong.data
```

```
 FILE DOES NOT EXIST

 WHICH DATA FILE IS TO RUN?

y11.data

 OUTPUT FOR THIS RUN WILL BE IN data   3

 ALL OUTPUT FILES ARE data.n, WHERE n MAY BE CHANGED

 DO YOU WISH TO CHANGE n? y or n

n

DO YOU WANT TO MAKE ANOTHER RUN? y or n

y

WHICH DATA FILE IS TO RUN?

y12.data

 OUTPUT FOR THIS RUN WILL BE IN data.   4

 ALL OUTPUT FILES ARE data.n, WHERE n MAY BE CHANGED

 DO YOU WISH TO CHANGE n? y or n

n

 DO YOU WANT TO MAKE ANOTHER RUN? y or n

n

 MASTER MENU

   1.  ACCESS BRISTOL RCI PROGRAM

   2.  ACCESS VSAS2 RCI PROGRAM

   3.  OPTIMUM ROUTE PROGRAM

   4.  EVALUATION OF RCI OVER A GIVEN ROUTE

   5.  FINISH

 INPUT OPTION 1-5

1
```

9.3 Run One VSAS2 File

```
MASTER MENU
   1. ACCESS BRISTOL RCI PROGRAM
   2. ACCESS VSAS2 RCI PROGRAM
   3. OPTIMUM ROUTE PROGRAM
   4. EVALUATION OF RCI OVER A GIVEN ROUTE
   5. FINISH
 INPUT OPTION 1-5
2
 MENU SELECTION
 1. CREATE A NEW DATA FILE
 2. CHANGE AN EXISTING DATA FILE
 3. RUN VSAS2
 4. EXIT WITHOUT A RUN
3
 INPUT NAME OF SEGMENT FILE
input.seg
 INPUT NAME OF STORM FILE
input.storm
 INPUT NAME OF INITIAL MOISTURE FILE
input.imc
 INPUT NAME OF SOIL DESCRIPTION FILE
input.ci
 PLEASE ENTER NAME OF OUTPUT FILE
data.7
```

```
MASTER MENU
   1. ACCESS BRISTOL RCI PROGRAM
   2. ACCESS VSAS2 RCI PROGRAM
   3. OPTIMUM ROUTE PROGRAM
   4. EVALUATION OF RCI OVER A GIVEN ROUTE
   5. FINISH
 INPUT OPTION 1-5
5
% (*back to system level*)
```

9.4a Evaluate a Given Route

```
master
 ****************************************************
 WELCOME TO THE BRISTOL ROUTE PACKAGE
 ****************************************************
 ENTER ANY NUMBER TO CONTINUE
4
 MASTER MENU
   1.  ACCESS BRISTOL RCI PROGRAM
   2.  ACCESS VSAS2 RCI PROGRAM
   3.  OPTIMUM ROUTE PROGRAM
   4.  EVALUATION OF RCI OVER A GIVEN ROUTE
   5.  FINISH
 INPUT OPTION 1-5
4
 PLEASE MAKE CHOICE OF  :-

  1.CREATE ROUTE
  2.READ ROUTE
1
 INPUT NUMBER OF NODES ON ROUTE
4
 INPUT DIST OF NODE ALONG ROUTE    1
2
 INPUT IDENTIFIER OF NODE    1
1
 INPUT TIME STEP OF NODE    1
3
```

```
 INPUT DIST OF NODE ALONG ROUTE  2
2
 INPUT IDENTIFIER OF NODE  2
4
 INPUT TIME STEP OF NODE   2
3
 INPUT DIST OF NODE ALONG ROUTE  3
2
 INPUT IDENTIFIER OF NODE  3
2
 INPUT TIME STEP OF NODE   3
4
 INPUT DIST OF NODE ALONG ROUTE  4
10
 INPUT IDENTIFIER OF NODE  4
3
 INPUT TIME STEP OF NODE   4
5
 INPUT NAME OF FILE ROUTE WILL BE SAVED ON
EVAL.ROUTE
 MASTER MENU
    1. ACCESS BRISTOL RCI PROGRAM
    2. ACCESS VSAS2 RCI PROGRAM
    3. OPTIMUM ROUTE PROGRAM
    4. EVALUATION OF RCI OVER A GIVEN ROUTE
    5. FINISH
 INPUT OPTION 1-5
5
```

9.4b Calculate Optimum Route

```
master
 ****************************************************
 WELCOME TO THE BRISTOL ROUTE PACKAGE
 ****************************************************
 ENTER ANY NUMBER TO CONTINUE
4
 MASTER MENU
   1. ACCESS BRISTOL RCI PROGRAM
   2. ACCESS VSAS2 RCI PROGRAM
   3. OPTIMUM ROUTE PROGRAM
   4. EVALUATION OF RCI OVER A GIVEN ROUTE
   5. FINISH
 INPUT OPTION 1-5
3
 *** MENU SELECTION FOR GRID SET-UP ***
  1. CREATE GRID
  2. CHECK GRID
  3. SAVE GRID
  4. READ EXISTING GRID
  5. EXIT TO ROUTE SELECTION
  6. QUIT PROGRAM

  PLEASE ENTER SELECTION 1-6
5
```

```
 IT IS ASSUMED THAT THERE ARE DATA FILES EXISTING

 NAMED data.n, WHERE n  IS THE ID IN THE GRID

 DO YOU WISH TO RETURN TO MAIN MENU?  y/n

n

 PLEASE INPUT TIME REQUIRED  (AS HOURS FROM START)

4

 PLEASE DEFINE MAX. NUMBER OF FILES TO BE ACCESSED

4

data.1

data.2

data.3

data.4

  INPUT NAME OF GRID FILE

input.grid

 ii   6

 jj  6


 DEFINE STATRTING POSITION


 ENTER COLUMN

4

 4


 ENTER ROW

2

 2
 DEFINE FINISH POSITION


 ENTER COLUMN

5
  5


 ENTER ROW

5
  5


 PLEASE ENTER NAME OF OUTPUT FILE

OP.ROUTE
 MASTER MENU
   1.  ACCESS BRISTOL RCI PROGRAM
   2.  ACCESS VSAS2 RCI PROGRAM
   3.  OPTIMUM ROUTE PROGRAM
   4.  EVALUATION OF RCI OVER A GIVEN ROUTE
   5.  FINISH
 INPUT OPTION 1-5
```

9.5 Edit One BSSS File and Run

```
master
 ************************************************
 WELCOME TO THE BRISTOL ROUTE PACKAGE
 ************************************************
 ENTER ANY NUMBER TO CONTINUE
2
 MASTER MENU
   1. ACCESS BRISTOL RCI PROGRAM
   2. ACCESS VSAS2 RCI PROGRAM
   3. OPTIMUM ROUTE PROGRAM
   4. EVALUATION OF RCI OVER A GIVEN ROUTE
   5. FINISH
 INPUT OPTION 1-5
1
 MENU SELECTION
  1. CREATE A NEW DATA FILE
  2  CHANGE AN EXISTING DATA FILE
  3. RUN BRISTOL SSS
  4. EXIT WITHOUT A RUN
2
 INPUT NAME OF FILE  TO BE READ
y10.data
```

```
PLEASE SELECT THE IDENTIFYING VALUE OF THE PARAMETER
YOU WISH TO CHANGE FROM THE FOLLOWING SERIES OF LISTS

1.METEOROLOGICAL DATA
2.PRECIPITATION DATA
3.SOILS DATA
4.RUN DEFINITION DATA
5.SAVE FILE
6.EXIT TO MAIN MENU
2
CHOOSE FROM THE FOLLOWING  PARAMETERS
1. NUMBER OF HOURS OF RAIN
2. ALL PRECIPITATION VALUES
3. ONE PRECIPITATION VALUE
4. RAIN START TIME
5. RAIN STOP TIME
6. EXIT TO MENU
4
INPUT RAIN START TIME
PRESENT VALUE IS   2
6
PLEASE SELECT THE IDENTIFYING VALUE OF THE PARAMETER
YOU WISH TO CHANGE FROM THE FOLLOWING SERIES OF LISTS

1.METEOROLOGICAL DATA
2.PRECIPITATION DATA
3.SOILS DATA
```

```
 4.RUN DEFINITION DATA

 5.SAVE FILE

 6.EXIT TO MAIN MENU

5

 WHAT NAME DO YOU WISH TO SAVE THIS FILE UNDER

new1.data

 INPUT ANY NUMBER TO CONTINUE

1

MENU SELECTION

 1.  CREATE A NEW DATA FILE

 2.  CHANGE AN EXISTING DATA FILE

 3.  RUN BRISTOL SSS

 4.  EXIT WITHOUT A RUN

3

 WHICH DATA FILE IS TO RUN?

new1.data

 OUTPUT FOR THIS RUN WILL BE IN data.   1

 ALL OUTPUT FILES ARE data.n, WHERE n MAY BE CHANGED

 DO YOU WISH TO CHANGE n? y or n

y

 INPUT VALUE FOR n 1-30

6

 DO YOU WANT TO MAKE ANOTHER RUN? y or n

n
```

```
MASTER MENU
  1. ACCESS BRISTOL RCI PROGRAM
  2. ACCESS VSAS2 RCI PROGRAM
  3. OPTIMUM ROUTE PROGRAM
  4. EVALUATION OF RCI OVER A GIVEN ROUTE
  5. FINISH
INPUT OPTION 1-5
5
%
```

9.6 Edit One VSAS2 Input File and Run

```
MASTER MENU
   1. ACCESS BRISTOL RCI PROGRAM
   2. ACCESS VSAS2 RCI PROGRAM
   3. OPTIMUM ROUTE PROGRAM
   4. EVALUATION OF RCI OVER A GIVEN ROUTE
   5. FINISH
 INPUT OPTION 1-5
2
 MENU SELECTION
  1. CREATE A NEW DATA FILE
  2. CHANGE AN EXISTING DATA FILE
  3. RUN VSAS2
  4. EXIT WITHOUT A RUN
2
 INPUT NAME OF SEGMENT FILE TO BE READ
seg.2
 INPUT STORM FILE NAME
storm.3
 INPUT INITIAL MOISTURE FILE NAME
imc.2
 INPUT SOIL DESCRIPTION FILE NAME
c1.2
 PLEASE SELECT THE IDENTIFYING VALUE OF THE PARAMETERS
 YOU WISH TO CHANGE FROM THE FOLLOWING SERIES OF LISTS
 1.METEOROLOGICAL DATA
 2.PRECIPITATION DATA
 3.SOILS DATA
 4.RUN DEFINITION DATA
 5.SAVE FILE
 6.EXIT TO MAIN MENU
1
 CHOOSE FROM THE FOLLOWING PARAMETERS
 1.PRECIPITATION VALUES
 2.COARSE ITERATION TIME INTERVAL
 3.FINE ITERATION TIME INTERVAL
 4.OPTION FOR PRINT OF SEGMENT GEOMETRY
 5.OPTION FOR CALC. OF 100% AND 75% SATURATED AREAS
 6.STONE CONTENT OF EACH LAYER
 7.SEGMENT NUMBER
 8.INCREMENTS ON SLOPE
 9.NUMBER OF SEGMENTS
 10.NUMBER OF SOIL LAYERS
 11.SEGMENT DIMENSIONS
 12.ROUTING DELAY
 13.SEGMENT DISTANCES
 14.***EXIT***
1
 CHOOSE FROM THE FOLLOWING PARAMETERS
 1.NUMBERS OF HOURS OF RAIN
 2.ALL PRECIPITATION VALUES
 3.ONE PRECIPITATION VALUE
 4.EXIT TO MENU
3
```

```
 WHICH HOUR OF PRECIPITATION DO YOU WISH TO ALTER?
10
 INPUT PRECIPITATION (m) FOR HOUR   10
 PRESENT VALUE IS   0.
0.01
 CHOOSE FROM THE FOLLOWING PARAMETERS
 1.NUMBERS OF HOURS OF RAIN
 2.ALL PRECIPITATION VALUES
 3.ONE PRECIPITATION VALUE
 4.EXIT TO MENU
3
 WHICH HOUR OF PRECIPITATION DO YOU WISH TO ALTER?
11
 INPUT PRECIPITATION (m) FOR HOUR   11
 PRESENT VALUE IS   0.
0.02
 CHOOSE FROM THE FOLLOWING PARAMETERS
 1.NUMBERS OF HOURS OF RAIN
 2.ALL PRECIPITATION VALUES
 3.ONE PRECIPITATION VALUE
 4.EXIT TO MENU
4
 PLEASE SELECT THE IDENTIFYING VALUE OF THE PARAMETERS
 YOU WISH TO CHANGE FROM THE FOLLOWING SERIES OF LISTS
 1.METEOROLOGICAL DATA
 2.PRECIPITATION DATA
 3.SOILS DATA
 4.RUN DEFINITION DATA
 5.SAVE FILE
 6.EXIT TO MAIN MENU
5
 INPUT NAME OF NEW SEGMENT FILE
seg.3
  FILE ALREADY EXISTS - DO YOU WISH TO OVERWRITE?


  y or n


y
 INPUT NAME OF NEW STORM FILE
storm.4
 INPUT NAME OF NEW INITIAL MOISTURE FILE
imc.3
 INPUT NAME OF NEW SOIL DESCRIPTION FILE
c1.4
  INPUT ANY NUMBER TO CONTINUE
3
 MENU SELECTION
 1.  CREATE A NEW DATA FILE
 2.  CHANGE AN EXISTING DATA FILE
 3.  RUN VSAS2
 4.  EXIT WITHOUT A RUN
```

```
3
 INPUT NAME OF SEGMENT FILE
seg.3
 INPUT NAME OF STORM FILE
storm.4
 INPUT NAME OF INITIAL MOISTURE FILE
imc.3
 INPUT NAME OF SOIL DESCRIPTION FILE
ci.4
 PLEASE INPUT NAME OF OUTPUT FILE
data.4


 FILE ALREADY EXISTS - DO YOU WISH TO OVERWRITE?


  y or n


n
 PLEASE INPUT NAME OF OUTPUT FILE
data.6
MASTER MENU
   1.  ACCESS BRISTOL RCI PROGRAM
   2.  ACCESS VSAS2 RCI PROGRAM
   3.  OPTIMUM ROUTE PROGRAM
   4.  EVALUATION OF RCI OVER A GIVEN ROUTE
   5.  FINISH
 INPUT OPTION 1-5
5
% (*back to system level*)
```

9.7 Edit a Grid to Incorporate New Files

```
master
 *************************************************
 WELCOME TO THE BRISTOL ROUTE PACKAGE
 *************************************************
 ENTER ANY NUMBER TO CONTINUE
4
 MASTER MENU
   1. ACCESS BRISTOL RCI PROGRAM
   2. ACCESS VSAS2 RCI PROGRAM
   3. OPTIMUM ROUTE PROGRAM
   4. EVALUATION OF RCI OVER A GIVEN ROUTE
   5. FINISH
 INPUT OPTION 1-5
3
 *** MENU SELECTION FOR GRID SET-UP ***
  1. CREATE GRID
  2. CHECK GRID
  3. SAVE GRID
  4. READ EXISTING GRID
  5. EXIT TO ROUTE SELECTION
  6. QUIT PROGRAM

  PLEASE ENTER SELECTION 1-6
1
  HOW MANY COLUMNS?
6
```

```
 HOW MANY ROWS ?
6
  FOR ROW    1  COLUMN    1  ENTER ID
1
  FOR ROW    1  COLUMN    2  ENTER ID
1
  FOR ROW    1  COLUMN    3  ENTER ID
1
  FOR ROW    1  COLUMN    4  ENTER ID
3
  FOR ROW    1  COLUMN    5  ENTER ID
3
  FOR ROW    1  COLUMN    6  ENTER ID
2
  FOR ROW    2  COLUMN    1  ENTER ID
1
  FOR ROW    2  COLUMN    1  ENTER ID
1
  FOR ROW    2  COLUMN    2  ENTER ID
2
  FOR ROW    2  COLUMN    3  ENTER ID
5
  FOR ROW    2  COLUMN    4  ENTER ID
4
  FOR ROW    2  COLUMN    5  ENTER ID
3
```

```
  FOR ROW    2  COLUMN    6  ENTER ID
2
  FOR ROW    3  COLUMN    1  ENTER ID
1
  FOR ROW    3  COLUMN    2  ENTER ID
2
  FOR ROW    3  COLUMN    3  ENTER ID
1
  FOR ROW    3  COLUMN    4  ENTER ID
1
  FOR ROW    3  COLUMN    5  ENTER ID
3
  FOR ROW    3  COLUMN    6  ENTER ID
3
  FOR ROW    4  COLUMN    1  ENTER ID
4
  FOR ROW    4  COLUMN    2  ENTER ID
6
  FOR ROW    4  COLUMN    3  ENTER ID
5
  FOR ROW    4  COLUMN    4  ENTER ID
4
  FOR ROW    4  COLUMN    5  ENTER ID
4
  FOR ROW    4  COLUMN    6  ENTER ID
3
```

```
FOR ROW    5  COLUMN    1  ENTER ID
2
FOR ROW    5  COLUMN    2  ENTER ID
3
FOR ROW    5  COLUMN    3  ENTER ID
3
FOR ROW    5  COLUMN    4  ENTER ID
3
FOR ROW    5  COLUMN    5  ENTER ID
2
FOR ROW    5  COLUMN    6  ENTER ID
3
FOR ROW    6  COLUMN    1  ENTER ID
1
FOR ROW    6  COLUMN    2  ENTER ID
1
FOR ROW    6  COLUMN    3  ENTER ID
3
FOR ROW    6  COLUMN    4  ENTER ID
2
FOR ROW    6  COLUMN    5  ENTER ID
3
FOR ROW    6  COLUMN    6  ENTER ID
3
```

```
            COLUMN

           1  2  3  4  5  6

ROW

 1         1  1  1  3  3  2

 2         1  2  5  4  3  2

 3         1  2  1  1  3  3

 4         4  6  5  4  4  3

 5         2  3  3  3  2  3

 6         1  1  3  2  3  3

*** MENU SELECTION FOR GRID SET-UP ***

1. CREATE GRID

2. CHECK GRID

3. SAVE GRID

4. READ EXISTING GRID

5. EXIT TO ROUTE SELECTION

6. QUIT PROGRAM


PLEASE ENTER SELECTION 1-6

2

            COLUMN

           1  2  3  4  5  6

ROW

 1         1  1  1  3  3  2

 2         1  2  5  4  3  2

 3         1  2  1  1  3  3

 4         4  6  5  4  4  3

 5         2  3  3  3  2  3

 6         1  1  3  2  3  3
```

```
DO YOU WISH TO CHANGE ANY GRID VALUE? Y or N
y
ENTER ROW NUMBER
3
ENTER COLUMN NUMBER
2
THE VALUE OF THIS CELL IS   2
PLEASE ENTER NEW IDENTIFYING VALUE
4
THE NEW GRID IS AS FOLLOWS


        COLUMN
         1  2  3  4  5  6
ROW
  1      1  1  1  3  3  2
  2      1  2  5  4  3  2
  3      1  4  1  1  3  3
  4      4  6  5  4  4  3
  5      2  3  3  3  2  3
  6      1  1  3  2  3  3

*** MENU SELECTION FOR GRID SET-UP ***
1.  CREATE GRID
2.  CHECK GRID
3.  SAVE GRID
4.  READ EXISTING GRID
5.  EXIT TO ROUTE SELECTION
6.  QUIT PROGRAM
PLEASE ENTER SELECTION 1-6
3
```

```
 DO YOU WISH TO SAVE THIS GRID?
y
 PLEASE TYPE IN NAME OF FILE THAT GRID IS TO
 BE SAVED IN
input.grid2
 *** MENU SELECTION FOR GRID SET-UP ***
 1.  CREATE GRID
 2.  CHECK GRID
 3.  SAVE GRID
 4.  READ EXISTING GRID
 5.  EXIT TO ROUTE SELECTION
 6.  QUIT PROGRAM
 PLEASE ENTER SELECTION 1-6
6
 MASTER MENU
   1.  ACCESS BRISTOL RCI PROGRAM
   2.  ACCESS VSAS2 RCI PROGRAM
   3.  OPTIMUM ROUTE PROGRAM
   4.  EVALUATION OF RCI OVER A GIVEN ROUTE
   5.  FINISH
  INPUT OPTION 1-5
5
% (*back to system level*)
```

9.8a Evaluate a Given Route

```
master
 *************************************************
 WELCOME TO THE BRISTOL ROUTE PACKAGE
 *************************************************
 ENTER ANY NUMBER TO CONTINUE
4
 MASTER MENU
   1.  ACCESS BRISTOL RCI PROGRAM
   2.  ACCESS VSAS2 RCI PROGRAM
   3.  OPTIMUM ROUTE PROGRAM
   4.  EVALUATION OF RCI OVER A GIVEN ROUTE
   5.  FINISH
 INPUT OPTION 1-5
4
 PLEASE MAKE CHOICE OF  :-


 1.CREATE ROUTE
 2.READ ROUTE
1
 INPUT NUMBER OF NODES ON ROUTE
4
 INPUT DIST OF NODE ALONG ROUTE    1
2
 INPUT IDENTIFIER OF NODE   1
1
 INPUT TIME STEP OF NODE   1
3
```

```
 INPUT DIST OF NODE ALONG ROUTE   2

2

  INPUT IDENTIFIER OF NODE   2

4

  INPUT TIME STEP OF NODE    2

3

 INPUT DIST OF NODE ALONG ROUTE   3

2

  INPUT IDENTIFIER OF NODE   3

2

  INPUT TIME STEP OF NODE    3

4

 INPUT DIST OF NODE ALONG ROUTE   4

10

  INPUT IDENTIFIER OF NODE   4

5

  INPUT TIME STEP OF NODE    4

5

  INPUT NAME OF FILE ROUTE WILL BE SAVED ON

EVAL.ROUTE2

 MASTER MENU

    1.  ACCESS BRISTOL RCI PROGRAM

    2.  ACCESS VSAS2 RCI PROGRAM

    3.  OPTIMUM ROUTE PROGRAM

    4.  EVALUATION OF RCI OVER A GIVEN ROUTE

    5.  FINISH

  INPUT OPTION 1-5

5

% (*back to system level*)
```

9.8b Calculate Optimum Route

```
master
 ************************************************
 WELCOME TO THE BRISTOL ROUTE PACKAGE
 ************************************************
 ENTER ANY NUMBER TO CONTINUE
4
 MASTER MENU
   1. ACCESS BRISTOL RCI PROGRAM
   2. ACCESS VSAS2 RCI PROGRAM
   3. OPTIMUM ROUTE PROGRAM
   4. EVALUATION OF RCI OVER A GIVEN ROUTE
   5. FINISH
 INPUT OPTION 1-5
3
 *** MENU SELECTION FOR GRID SET-UP ***
 1. CREATE GRID
 2. CHECK GRID
 3. SAVE GRID
 4. READ EXISTING GRID
 5. EXIT TO ROUTE SELECTION
 6. QUIT PROGRAM

 PLEASE ENTER SELECTION 1-6
5
```

```
  IT IS ASSUMED THAT THERE ARE DATA FILES EXISTING
  NAMED data.n, WHERE n  IS THE ID IN THE GRID
  DO YOU WISH TO RETURN TO MAIN MENU?  y/n
n
  PLEASE INPUT TIME REQUIRED  (AS HOURS FROM START)
4
  PLEASE DEFINE MAX. NUMBER OF FILES TO BE ACCESSED
6
 data.1
 data.2
 data.3
 data.4
 data.5
 data.6
   INPUT NAME OF GRID FILE
input.grid2
   ii  6
   jj 6

  DEFINE STATRTING POSITION

  ENTER COLUMN
 4
   4

  ENTER ROW
 2
   2
 DEFINE FINISH POSITION


 ENTER COLUMN
5
  5


 ENTER ROW
5
  5


 PLEASE ENTER NAME OF OUTPUT FILE
OP.ROUTE2
 MASTER MENU
    1. ACCESS BRISTOL RCI PROGRAM
    2. ACCESS VSAS2 RCI PROGRAM
    3. OPTIMUM ROUTE PROGRAM
    4. EVALUATION OF RCI OVER A GIVEN ROUTE
    5. FINISH
 INPUT OPTION 1-5
```

10 DISCUSSION

10.1 Development and Completion of Initial BORTS Software

Throughout the development of the BSSS and route management schemes, and the adaptation of the VSAS2, the strategy described in **section 1.4** has been utilized. The resulting 'Bristol Off-Road Trafficability Scheme' (BORTS) can be considered as a first attempt to fulfil the requirements set out in **section1.2** and hence provide a strong framework for further development.

The simulation of the physical environmental processes and the modelling of the complex world through simplified representation has introduced restrictions and limitations in the application of the scheme. These restrictions and limitations have been outlined in **section 8**. The further development of the scheme may or may not reduce the effect of these restrictions and limitations, but it is more likely that they will be replaced by a different set. With this view in mind the developer must be careful not to 'improve' the scheme at the expense of a requirement. This is the stage where knowledge of the subject area is important so that a qualitative assessment of the requirements and their relative comparative importance can be made.

After examination of the BORTS, it will be apparent that there are several areas where there is scope for 'improvement' and development. Before specifying these areas it is important to assess the first generation scheme so that a detailed prescription for future development can be written. It is also important to assess the performance of the BSSS, VSAS2 and the SMSP under different conditions so that a guide to when the use of one scheme is more appropriate than another.

The following sections attempts to indicate the different computational requirements of the BSSS and the VSAS2, and the different features of the output from the BSSS, VSAS2 and the SMSP schemes and hence make some recommendations for application.

10.2 Computational Requirements of the BSSS and the VSAS2

The computational requirements of the BSSS and the VSAS2 schemes are very different. **Table 10.1** shows some results of computer use versus iteration time interval and duration of run for the VSAS2 and BSSS on a systime 8750 with a UNIX operating system. The results were obtained using the 'time' command while calling 'master', i.e.

```
time master
```

The appropriate scheme was then run according to **sections 9.1** and **9.3** for the BSSS and the VSAS2 respectively. The times given in **table 10.1** are in seconds and the iteration specifications for each scheme are given in the manner required. That is, the BSSS requires the iteration time step in seconds, whereas, the VSAS2 requires a coarse iteration specification (krep) which refers to the number of times within an hour that calculations are made, i.e. when krep = 4, the variables are all recalculated every 15 minutes. The fine scale iteration (lrep), used when near saturation occurs, refers to the number of times within the coarse scale iteration that recalculation of the variables takes place, i.e., when krep = 4 and lrep = 3, the iteration interval for areas of near saturation will be 5 minutes. It is evident from the fact that the iteration interval for the

TABLE 10.1 COMPARISON OF COMPUTING REQUIREMENTS

BSSS VERSUS VSAS2

(Systime 8750 under UNIX)

VSAS2

Duration of Run	KREP	LREP	TIME1	TIME2
32	2	2	38.75	4.3
32	4	3	89.3	4.5
32	6	3	130.3	5.7
32	12	2	171.8	7.8
32	12	5	406.8	12.9
80	2	2	95.9	8.9
80	4	3	233.2	14.5

BSSS

Duration of Run (hours)	ITERATION TIME STEP (secs)	TIME1	TIME2
32	120	41.0	0.8
32	60	78.3	1.55
80	300	45.0	1.7
80	150	85.1	0.8

KEY: TIME1 = time taken executing the commands (seconds)

TIME2 = time taken searching for files (seconds)

VSAS2 scheme can vary both temporally and spatially that the times taken to run the scheme will vary according to initial moisture content, soil hydrological characteristics, topography and the rain storm characteristics. The results given in **table 10.1** represent an area of moderate slope (approximately 7 degrees) which starts off with no area saturated but where the lower quarter of the slope becomes nearly saturated between hours 15 and 25.

Although the results shown in **table 10.1** cannot be viewed as absolute, they do provide some guide as to the relative computer usage of the two schemes. As expected with both schemes, the time taken to execute the commands (TIME1) increases as the interval between calculations decreases. TIME1 also increases with an increase in run time.

The time spent on the system, including the time taken searching for files (TIME2) is small in comparison to TIME1, but there is a significant difference between the VSAS2 and the BSSS. The VSAS2 requires much more TIME1 in comparison to the BSSS. This might be accounted for by the increased number of input files (four for the VSAS2, whereas only one for the BSSS), and the extra output still retained for the VSAS2 for development purposes.

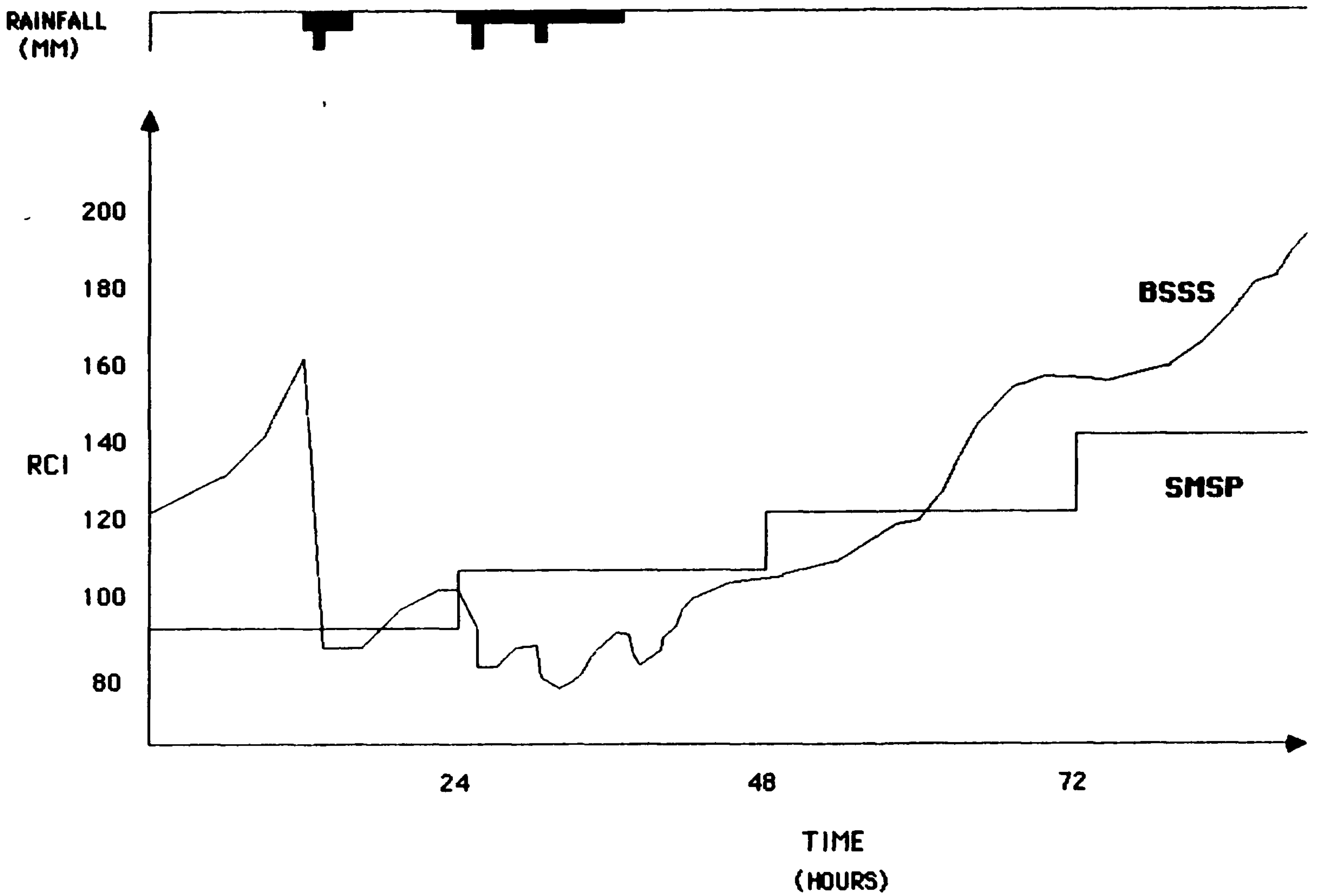
The information given in **table 10.1** indicates the significant difference in TIME1 values between the schemes. Consider a duration of eighty hours for each scheme and an iteration scheme of 5 minutes (300 seconds) or where krep = 4 and lrep = 3. Although the VSAS2 scheme is not operating at this fine a scale for the whole duration or for the whole segment area, the TIME1 value of 233 seconds as compared to the BSSS TIME1 value of 45 seconds is approximately five times greater. This indicates a potential restriction for the use of the VSAS2 as compared to the BSSS. The question which must also be taken into consideration is "at what iteration time intervals do both schemes become potentially unstable?" This will be an important aspect of running the two schemes. For the purposes of this analysis the information given in **table 10.1** allows us to observe the manner in which the computing time is influenced by iteration time intervals and so set out further guidelines (**section 10.4**) for the utilization of the VSAS2 and BSSS within the BORTS.

10.3 Initial Results BSSS versus SMSP

Initial runs of the BSSS have been made for comparison with runs of the SMSP. **Figure 10.1** shows the nature inwhich the BSSS and the SMSP systems differ in predictive resolution. That is, the SMSP is only able to predict a daily RCI value and so appears as a series of well defined steps. Alternatively, the BSSS shows the continuous nature of the processes affecting the soil surface layers, i.e. the processes of drainage, evaporation and rainfall in a series of much smaller time steps and hence approximating this continuous nature much closer than the SMSP.

In **section 10.2** the computational requirements of the VSAS2 and the BSSS were discussed. Allowing for the preparation time required for each of the three scheme, i.e BSSS, VSAS2 and SMSP, it is the actual computational requirements that will be considered. The computational requirements of the SMSP are not available but are being assumed to almost negligible as the operation of the SMSP is that of mainly look-up tables. The preparation time for each scheme will vary according to information available, complexity of the situation and the familiarity and expertise of the user.

FIGURE 10.1 INITIAL RESULTS FROM SMSP AND BSSS MODELS



10.4 Use of Different Schemes

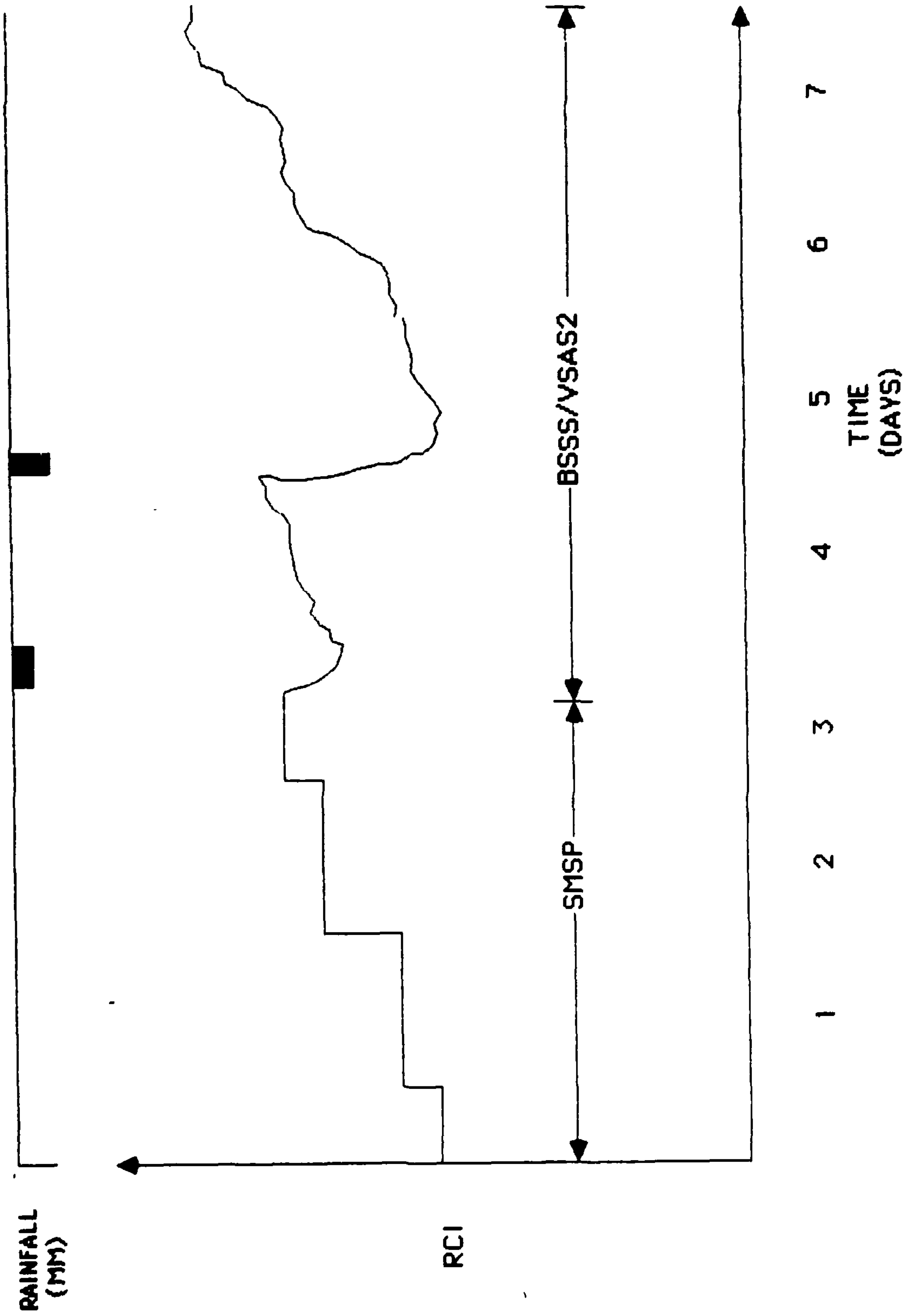
The above sections have shown that there are three potential schemes for the prediction of RCI. The SMSP is the existing scheme with the BSSS and the VSAS2 as the new schemes. The three schemes have their own set of limitations and advantages over the others and it is important to formulate some framework whereby the user may assess which scheme would be the most appropriate to use under different circumstances.

The choice of model will mainly be dependent on the following factors:-

(a) computational time available,
(b) storm characteristics,
(c) initial moisture content,
(d) soil type,
(e) time of day and year,
(f) topography,
(g) spatial resolution, and
(h) temporal resolution.

The relative importance of the computational restrictions is going to be a function of the machine, its load and any time restrictions imposed by the user. Spatial and temporal resolution will also affect the choice of scheme, for instance, if information is required for an area over the next six hours the SMSP will be unable to provide information at this resolution. On the other hand if information about conditions three days ahead is required on a daily basis the SMSP would be the most appropriate scheme to use. AS shown in **figure 10.2**, it might be desirable to run the

FIGURE 10.2 EXAMPLE OF SCHEMES RUNNING IN TANDEM



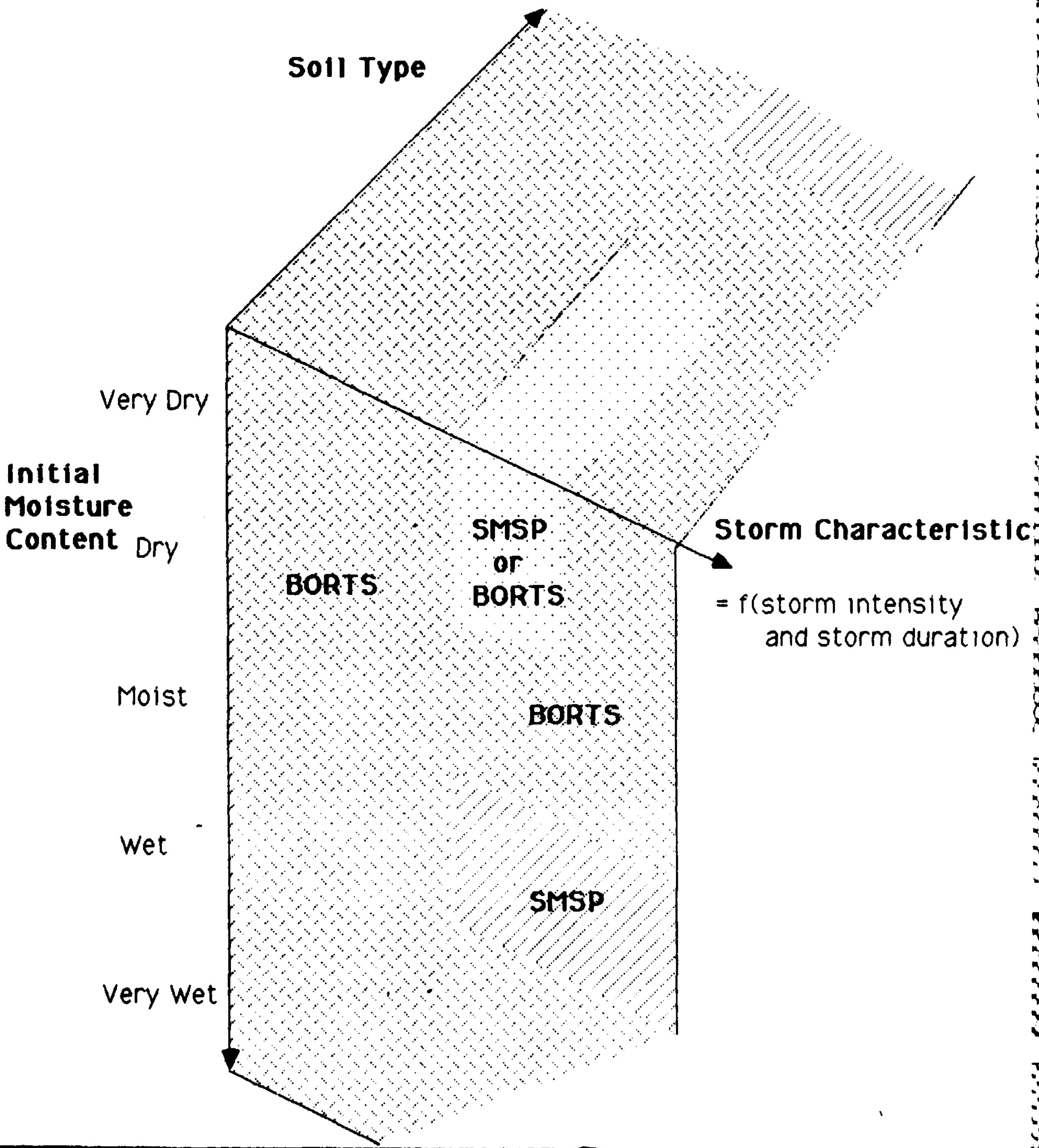
SMSP and the BSSS or VSAS2 in tandem, i.e., run the SMSP for a number of days to determine the approximate initial moisture content and then run either the BSSS or the VSAS2 for the shorter time period where greater temporal resolution is required.

The manner inwhich the storm characteristics, soil type and initial moisture content might affect the choice is shown in **figure 10.3**. Runs from the SMSP with initial moisture contents of very wet and very dry have been excluded from the choice procedure as the results indicated some peculiarities in the model, i.e very wet conditions never recovered and very dry conditions were almost unaffected by even heavy rain conditions.

The development of a guide to model choice as shown in **figure 10.3** is currently under way but it should be remembered that all the factors listed above will be relevant in the choice of the most appropriate model. The 'weighting' of these factors must at present be carried out by the user and although having a framework such as **figure 10.3** is very helpful, especially for the inexperienced user, some experience in using the schemes will enable an easier choice to be made.

Use of the different schemes for various conditions over an area would enable the set-up of a databank providing a scenario of results as a design table. The various "keys" for the data bank would of course be the factors **a-h** listed above. Once set up, this would require less expertise to use, fewer decisions and would for areas of interest, be a considerable time and hence money saver.

FIGURE 10.3 CHOICE OF DIFFERENT MODELS USING THREE FACTORS



10.5 Prescription for Further Development

Through examination of the limitations and restrictions of the major components of the BSSS and its preliminary comparison with the SMSP model it is possible to set out possible areas for further development:-

(a) Inclusion of evaporation within the VSAS2.

(b) Inclusion of the effect of vegetation on the hydrology in both the BSSS and VSAS2 schemes.

(c) Development of 'results data bank' providing a scenario of results as a design tool.

(d) Inclusion of automatic calculation of speed from RCI via DRIVEB model in the route management scheme.

(e) Assessment of a three dimensional soil water scheme for modelling small specific areas.

(f) Further development of topographic influence on initial moisture contents, hydrological and soil characteristics as an automatic look-up procedure.

(g) Application to plant growth problems, agricultural 'workability' of the land etc.

(h) Application to construction trails in terms of trafficability and slope stability.

(i) It is realised that there will be a variety of situations where the existing SMSP model will be more appropriate than either the BSSS or the VSAS2 scheme. On the other hand there are many situations when the BSSS or the VSAS2 will be more appropriate. The use of the schemes in tandem, as discussed in **section 10.3**, could provide a very powerful modelling tool. This type of utility guide requires immediate specification.

REFERENCES

Anderson, M.G., (1983) 'Forecasting off-road trafficability', Applied Geography, 3, 239-253.

Anderson, M.G. and J.E. Cochrane, (1987) 'Evaluation of deterministic models for near surface soil moisture prediction', U.S. Army European Research Office, London, Fifth Technical Report, DAJA45-85-C-0007, 88pp.

Baier, W., et al., (1979) 'The versatile soil moisture budget', Canadian Journal of Plant Science, 46, 299-315.

Ballick, M., L.E. Link, R.K. Scoggins and J.L. Solomon, (1981) 'Thermal modelling of terrain surface elements', U.S. Army Waterways Experimental Station, Technical report E1-81-2.

Bernier, P.Y., (1982) 'VSAS2: A revised source area simulator for small forested basins' Unpublished Ph.D. Thesis, University of Reading.

Bouwer, H., (1966) 'Rapid field measurement of air-entry value and hydraulic conductivity of soil as significant parameters in flow system analysis', Water Resour. Res. 2, 729-738.

Brakensiek, D.L., (1977) 'Estimating the effective capillary pressure in the Green and Ampt infiltration equation', Water Resour. Res. 13(3), 680-682.

Brakensiek, D.L. and C. Onstad, (1977) 'Parameter estimation of the Green and Ampt infiltration equation', Water Resour. Res. 13(6), 1009-1012.

Brakensiek, D.L. and W.J. Rawls, (1983) 'Use of infiltration procedures for estimating runoff', paper presented to the United States Department of Agriculture Soil Conservation Service, National Engineering Workshop, Tempe, AZ, USA.

Brooks, R.H. and A.T. Corey, (1964) 'Hydraulic properties of porous media', Hydrology Paper No. 3., Colorado State University, Fort Collins.

Campbell, G.S., (1974) 'A simple method for determining unsaturated conductivity from moisture retention data', Journal of Soil Science, 117(6), 311-314.

Childs, E.C. and N. Collis-George, (1950) 'The permeability of porous materials', Proc. R. Soc. London Ser. A 201, 392-405.

Childs, E.C., (1969) 'An introduction to the physical basis of soil water phenomena', Wiley, London.

Chu, S.T., (1978) 'Infiltration during unsteady rain', Water Resour. Res. 14(3), 461-466.

Clapp, R.B., (1982) 'A wetting-front model of soil water dynamics', Unpublished Ph.D. Thesis, Department of Environmental Sciences, University of Virginia.

Collins, J.G., (1967) 'Influences of water tables on soil moisture and soil strength', in 'Report of conference on soil trafficability prediction' Appendix D, U.S. Army Waterways Experiment Station.

Collins, J.G., (1971) 'Forecasting trafficability', U.S. Army Waterways Experiment Station, Technical Memorandum 3-331, Report No. 10.

Darcy, H., (1856) 'Les fontaines publique de la ville de Dijon', Dalmont, Paris.

Department of the Army, (1959) 'Soils trafficability', Technical Bulletin No. TB ENG 37 (AGO 166A - July).

Dyer, J.A. et al., (1978) 'Spring field workday probabilities for selected sites across Canada', Agrometeorology Section, Research Branch, Agriculture Canada, Ottawa, Ontario, Technical Bulletin 86.

Dyer, J.A., (1980) 'Fall field workdays in Canada', Agrometeorology Section, Research Branch, Agriculture Canada, Ottawa, Ontario, Technical Bulletin 92.

Dyer, J.A. and A.R. Mack, (1984) 'The versatile soil moisture budget - version three', Agrometeorology Section, Research Branch, Agriculture Canada, Ottawa, Ontario, Technical Bulletin No. 1984-1 E, Contribution No. 82-33.

Gardner, W.R., (1958) 'Some steady state solutions of the unsaturated moisture flow equation with application to evaporation from a water table', Soil Sci. 85(4).

Ghosh, R.K., (1977) 'Determination of unsaturated hydraulic conductivity from moisture retention function', Soil Sci. 124(2), 123-124.

Green, W.H. and G.A. Ampt, (1911) 'Studies on soil physics:I Flow of air and water through soils', J. Agr. Sci. 4, 1-24.

Green, W.H. and J.C. Corey, (1971) 'Calculation of hydraulic conductivity: A further evaluation of predictive methods', Soil Sci. Soc. Am. Proc. 35, 3-8.

Gupta, S.C. and W.E. Larson, (1979) 'Estimating soil water retention characteristics from particle size distribution, organic matter percent and bulk density', Water Resour. Res. 15(6),1630-1635.

Hanl, G. and P. Tries, (1982) 'A method for evaluating terrain trafficability of tanks with respect to weather influences', German Military Geophysical Office.

Haurwitz, B., (1948) 'Insolation in relation to cloud types', Journal of Meteorology, 5, 1673-1680

Henderson-Sellers, A., (1986) Personal Communication.

Hillel, D., (1982) 'Introduction to soil physics', Academic Press.

Holmes, R.M. and G.W. Robertson, (1959) 'A modulated soil moisture budget', Monthly Weather Review, 87, 101-106.

Holtan, H.N., (1961) 'A concept for infiltration estimates in watershed engineering', U.S. Department of Agriculture, Agricultural Research Service Publication 41-51.

Horton, R.E., (1940) 'An approach toward a physical interpretation of infiltration-capacity', Soil Sci. Soc. Am. Proc. 5, 399-417.

Hubbert, M.K., (1956) 'Darcy's law and the field equations of the flow of underground fluids', Am. Inst. Min. Met. Petl. Eng. Trans. 207, 222-239.

Huck, M.G. et al., (1975) ' Leaf water potential and moisture balance - field data', American Society of Agricultural Engineers, paper no. 75-2582.

Jackson, R.D. et al., (1965) 'Heat transfer' in "Methods of soil analysis", Monograph No. 9, Am Soc. Agron., Madison, WI., 349-360.

Jackson, R.D., (1972) 'On the calculation of hydraulic conductivity', Soil Sci. Soc. Am. Proc. 36, 380-382.

James, L.G. and C.L. Larson, (1976) 'Modelling infiltration and redistribution of soil water during intermittent application', Trans. ASAE 19(3), 482-488.

Kasten, F. and E. Raschke, (1974) 'Reflection and Transmission Terminology by analogy with scattering', Appl. Opt., 13, 460-464.

Khale, A.B., (1977) 'A simple model of the Earth's surface for geologic mapping by remote sensing', Journal of Geophysical Research, 82, 1673-1680.

Kriebel, K.T., (1977) 'Reflection properties of vegetated surfaces: Tables of measured spectral biconical reflectance factors', Meteorology Institute, University of Munich, Report No. 29.

Kunze, R.J. et al., (1968) 'Factors important in the calculation of hydraulic conductivity', Soil Sci. Soc. Am. Proc. 32, 760-765.

Marshall, T.J., (1958) 'A relation between permeability and size distribution of pores', J. Soil Sci. 9, 1-8.

Mein, R.G. and C.L. Larson, (1973) 'Modelling infiltration during a steady rain', Water Resour. Res. 9(2), 384-394.

Millington, R.J. and J.P. Quirk, (1959) 'Permeability of porous media', Nature, 183, 387-388.

Moltham, H.D., (1967) 'Influence of soil variability on soil moisture and soil strength predictions', in 'Report of conference on soil trafficability prediction' Appendix E. U.S. Army Waterways Experiment Station.

Monteith, J.L. and G. Szeicz, (1961) 'The radiation balance of bare soil and vegetation', Quarterly Journal Royal Meteorological Society, 87, 159-170.

Murry, F.W., (1967) 'On the computation of saturated vapor pressure', Journal of Applied Meteorology, 6, 203-204.

Nicodemus, F.E., (1970) "Reflectance nomenclature and directional reflectance and emissivity', Appl. Opt. 9, 1474-1475.

Nielsen, D.R. et al., (1973) 'Spatial variability of field-measured soil water properties', Hilgardia 42, 215-259.

Oke, T.R., (1978) 'Coundary layer climates', Methven and Co. Ltd., Halsted Press Book, John Wiley and Sons, N.Y., 322-334.

Philip, J.R., (1957) 'The theory of infiltration: 1. The infiltration equation and its solution', Soil Sci. 83, 345-357.

Pochop, L.O., M.D. Shanklin and D.A. Horner, (1968) 'Sky cover influence on total hemisphere radiation during daylight hours', Journal of Applied Meteorology, 7, 484-489.

Rawls, W.J., D.L. Brakensiek and K.E. Saxon, (1982) 'Estimation of soil water properties', Transactions of the American Society of Agricultural Engineers, 25(5), 1316-1320.

Reeves, M. and E.E. Miller, (1975) 'Estimating infiltration for erratic rainfall', Water Resour. Res. 11(1), 102-110.

Rhavagan, G.S.V. and F. McKyes, (1978) 'Traffic - soil - plant (maize) relations', Journal of Terramechanics, 16, 181-189.

Rohani, B. and G.Y. Baladi, (1981) 'Correlation of mobility cone index with fundamental engineering properties of soil', U.S. Army Waterways Experiment Station, Miscellaneous Paper SL-81-4.

Sellers, W.D., (1965) 'Physical Climatology', University of Chicago, Illinois.

Smith, M.H. and M.P. Meyer, (1973) 'Automation of a model for predicting soil moisture and soil strength (SMSP model)', U.S. Army Waterways Experiment Station, Miscellaneous Paper M-73-1.

Smith, W.L., (1966) 'Note on the relationship between total precipital water and surface dew point', Journal of Applied Meteorology, 5, 726-727.

Swartzendruber, D., (1969) 'The flow of water in unsaturated soils' in "Flow through porous media", editor R.J.M. DeWiest, chapter 6, 215-292.

Taylor, H.M., (1971) 'Effect of soil strength on seedling emergence, root growth and crop yield', in 'Compaction of Agricultural Soils, K.K. Barnes (editor), American Society of Agricultural Engineers Monograph, 292-305.

Thornthwaite, C.W. and J.R. Mather, (1955) 'The water balance', Climatology VIII(1):Drexel Institute of Technology, N.J.

U.S. Army Engineer Waterways Experiment Station, CE, 'Report of conference on soil trafficability prediction,' Apr 1967, Vicksburg, Mississippi, U.S.A.

Whitelaw, A., (1987) Personal Communication on VSAS2, University of Bristol, England.

Whitelaw, A., (1988) ' VSAS3', Unpublished Ph.D. Thesis, University of Bristol, England.

Youngs, E.G., (1968) 'An estimation of sorptivity for infiltration studies from moisture movement considerations', Soil Sci. 106, 157-163.

Appendix A : Code Description and Glossary

A1 : BSSS

The following section contains a list of the parameters and variables used in the BSSS. It is arranged in the alphabetical order of the variable name. Each name is accompanied by the common block they may be in, the subroutines they are used in and a brief description of their function.

Variable name: **a1**
Subroutines: 'bcone'
Description: intermediate calculation

Variable name: **a2**
Subroutines: 'bcone'
Description: intermediate calculation

Variable name: **a3**
Subroutines: 'bcone'
Description: intermediate calculation

Variable name: **aa1**
Common blocks: 'spec.bthree'
Subroutinēs: 'bcone', 'bread'
Description: cone length

Variable name: **aax**(j,i)

Common blocks: 'spec.bman'

Subroutines: 'bcntl', 'bsoilbd'

Description: θ values calculated in 'bsoilbd' for use in the 10 point

$\theta-\psi$ curve

Variable name: **abp**(11)

Common blocks: 'spec.bnow', 'spec.bd1'

Subroutines: 'bschars'

Description: mean bubbling pressure (air entry matric potential) for soils 1..11

Variable name: **acp**(11)

Common blocks: 'spec.bnow', 'spec.bd1'

Subroutines: 'bschars'

Description: mean of clay content percentage distribution for soil types 1...11

Variable name: **af**

Common blocks: 'spec.btwo'

Subroutines: 'binfil', 'bmenu', 'bread', 'bview'

Description: iteration time interval (seconds)

Variable name: **alb**

Common blocks: 'spec.bthree'

Subroutines: 'bevapor', 'bmenu', 'bread', 'bview'

Description: albedo of surface

Variable name: **alpha**

Common blocks: 'spec.bthree'

Subroutines: 'bcone', 'bread'

Description: cone tip half angle

Variable name: **alphi**

Subroutines: 'bevapor'

Description: thermal diffusivity

Variable name: **alr**

Common blocks: 'spec.bthree'

Subroutines: 'binfil', 'bmenu', 'bread', 'bview'

Description: rain start time

Variable name: **am**(11)

Common blocks: 'spec.bnow', 'spec.bd1'

Subroutines: bschars'

Description: mean of mineral bulk density distribution of soil types 1..11

Variable name: **amr**

Common blocks: 'spec.bthree'

Subroutines: 'binfil', 'bmenu', 'bread', 'bview'

Description: rain stop time

Variable name: **ans**

Description: character variable referring to an answer from the user
to a system prompt (y or n)

Variable name: **answ**

Description: character variable set to 'y' or 'n' by the user in response to a system query

Variable name: **aos**

Common blocks: 'spec.bgrnd'

Subroutines: 'bmenu', 'bread', 'bschars', 'bview'

Description: topographic index

Variable name: **arg1**

Subroutines: 'bevapor'

Description: energy contributed by atmospheric IR

Variable name: **asatcon**(11)

Common blocks: 'spec.bnow', 'spec.bdata'

Subroutines: 'bschars'

Description: mean saturated hydraulic conductivity for soils 1...11

Variable name: **asp**(11)

Common blocks: 'spec.bnow', 'spec.bd1'

Subroutines: 'bschars'

Description: mean of sand content percentage distribution of soil types 1..11

Variable name: **asr**(11)

Common blocks: 'spec.bnow', 'spec.bd1'

Subroutines: 'bschars'

Description: mean saturation for soils 1..11

Variable name: **ax**(j,i)

Common blocks: 'spec.bman'

Subroutines: 'bcntl', 'bsmcurv'

Description: θ values in the 10 point $\theta - \psi$ curve passed to 'bsmcurv'

Variable name: **b**

Subroutines: 'bevapor'

Description: heat conductivity of surface

Variable name: **b1**

Subroutines: 'bcone'

Description: intermediate calculation

Variable name: **b2**

Subroutines: 'bcone'

Description: intermediate calculation

Variable name: **b3**

Subroutines: 'bcone'

Description: intermediate calculation

Variable name: **b4**

Subroutines: 'bcone'

Description: intermediate calculation

Variable name: **bbb**(j,i)

Common blocks: 'spec.bfour'

Subroutines: 'balter', 'bevapor'

Description: base line for observations

Variable name: **bdm**(11)

Common blocks: 'spec.bgrnd'

Subroutines: 'bschars', 'bsoilbd'

Description: mineral bulk of each soil layer

Variable name: **bfile**(30)

Common blocks: 'spec.bgrnd'

Subroutines: 'bcntl', 'binfil'

Description: character array to keep the name of the output files in a logical order, but the name of the output file 'data.n' might not have the same numeric identifier - e.g miss(1) may be 'data.4' and miss(2) may be 'data.5' when files 'data.1' - 'data.3' have already been calculated

Variable name: **bots**

Subroutines: 'bhydcon'

Description: lower point on 20 point θ-k curve that observation is next to

Variable name: **bp**(j)

Common blocks: 'spec.bgrnd'

Subroutines: 'bcntl', 'bschars'

Description: air entry matric potential (bubbling pressure)

Variable name: **bprm**

Subroutines: 'bevapor'

Description: heat conductivity of bottom boundary layer

Variable name: **bsatcon**

Subroutines: 'bschars'

Description: intermediate calulation

Variable name: **bterm**

Subroutines: 'bevapor'

Description: energy contributed by insolation after adjustment using surface absorptivity

Variable name: **c**

Common blocks: 'spec.bthree'

Subroutines: 'bci', 'bmenu', 'bread', 'bview'

Description: soil cohesion

Variable name: **ci**

Subroutines: 'bci'

Description: cone index

Variable name: **cp**(11)

Common blocks: 'spec.bgrnd'

Subroutines: 'bschars', 'bsoilbd'

Description: clay content percentage for each soil layer

Variable name: **d**

Common blocks: 'spec.bthree'

Subroutines: 'bcone', 'bread'

Description: cone diameter

Variable name: **d1**

Subroutines: 'bcone'

Description: intermediate calculation

Variable name: **d2**

Subroutines: 'bcone'

Description: intermediate calculation

Variable name: **d3**

Subroutines: 'bcone'

Description: intermediate calculation

Variable name: **day**

Subroutines 'spec.btwo'

Subroutines: 'bcntl', 'bevapor', 'binfil'

Description: julian day

Variable name: **decl**

Subroutines: 'bevapor'

Description: solar declination angle

Variable name: **delt**

Subroutines: 'bevapor'

Description: time step in hours

Variable name: **dterm**

Subroutines: 'bevapor'

Description: energy loss due to evaporation

Variable name: **e1**

Common blocks: 'spec.bout'

Subroutines: 'bci', 'bcone'

Description: intermediate calculation passed from 'bcone'

Variable name: **e2**

Common blocks: 'spec.bout'

Subroutines: 'bci', 'bcone'

Description: intermediate calculation passed from 'bcone'

Variable name: **e4**

Common blocks: 'spec.bout'

Subroutines: 'bci', 'bcone'

Description: intermediate calculation passed from 'bcone'

Variable name: **elf**

Subroutines: 'bevapor'

Description: latitude in radians

Variable name: **epsn**

Subroutines: 'bevapor'

Description: emissivity of surface

Variable name: **facth**

Subroutines: 'bevapor'

Description: intermediate calculation in solving hterm

Variable name: **fexist**

Description: logical test variable used to test if a file exists

Variable name: **fmm**(j,i)

Common blocks: 'spec.bfour'

Subroutines: 'balter', 'bevapor'

Description: rate of change of met. observations

Variable name: **gamma**

Common blocks: 'spec.bthree'

Subroutines: 'bcone', 'bread'

Description: soil density

Variable name: **gg**

Common blocks: 'spec.bthree'

Subroutines: 'bci', 'bmenu', 'bread', 'bview'

Description: soil shear modulus

Variable name: **hterm**

Subroutines: 'bevapor'

Description: energy loss or gain due to convection

Variable name: **i**

Description: counter

Variable name: **icount**

Subroutines: 'bcntl'

Description: counter

Variable name: **ii**

Description: counter set to an menu option number by the user

Variable name: **iij**

Subroutines: 'bhydcon'

Description: counter on the 20 point soil - moisture characteristic curve

Variable name: **ij**

Subroutines: 'bevapor'

Description: counter

Variable name: **il**

Subroutines: 'bcntl'

Description: counter for number of runs to be made

Variable name: **imax**

Subroutine: 'balter'

Description: counter to ensure that the rate of change of the above varibles is only calculated between observations

Variable name: **immax**

Common blocks: 'spec.bthree"

Subroutines: 'binfil', 'bmenu', 'bread', 'bview'

Description: duration of run time

Variable name: **ir**

Common blocks: 'spec.btwo"

Subroutines: 'binfil', 'bmenu', 'bread', 'bview'

Description: number of hours of rain

Variable name: **iset**

Common block: 'bgrnd'

Subroutines: 'bcntl', 'binfil', 'bmenu'

Description: check variable - checks if input data file has already been read

Variable name: **isty**(10)

Common blocks: 'spec.bgrnd'

Subroutines: 'bmenu', 'bread', 'bschars', 'bview'

Description: soil type

Variable name: **ival**

Subroutines: 'bcntl'

Description: 'identifier' used in 'data.n', where n = ival

Variable name: **j**

Description: counter

Variable name: **jr**

Description: test variable - checking if the user is editing or creating an input file

Variable name: **jt**

Subroutines: 'bhydcon'

Description: counter on the 20 point soil - moisture characteristic curve

Variable name: **k**

Subroutines: 'bmenu', 'bread', 'bview'

Description: counter

Variable name: **k1**

Subroutine: 'bcntl', 'bmenu'

Description: sets input - output status

Variable name: **lat**

Common blocks: 'spec.btwo'

Subroutines: 'bevapor', 'bmenu', 'bread', 'bview'

Description: latitude of site

Variable name: **m**

Subroutines: 'bevapor'

Description: secant of solar zenith angle in radians

Variable name: **m**

Subroutines: 'bcone'

Description: intermediate calculation

Variable name: **max**(6)

Common blocks: 'spec.btwo'

Subroutines: 'bevapor', 'bmenu', 'bread', 'bview'

Description: number of met. observations (alg. 20211d)

Variable name: **miss**(30)

Subroutines; 'bcntl'

Description: name of input data file

Variable name: **name**

Description: character variable set to name of file

Variable name: **ncloud**

Common blocks: 'spec.btwo'

Subroutines: 'bevapor', 'bmenu', 'bread', 'bview'

Description: cloud type

Variable name: **nla**(10)

Common blocks: 'spec.bthree'

Subroutines: 'binfil', 'bmenu', 'bread', 'bview'

Description: number of cells in soil profile

Variable name: **nos**

Common blocks: 'spec.bone'

Subroutines: 'bcntl', 'bmenu', 'bread', 'bview'

Description: number of soil layers

Variable name: **ntabl**

Subroutines: 'bevapor'

Description: table number

Variable name: **op**(11)

Common blocks: 'spec.bone'

Subroutines: 'bmenu', 'bread', 'bsoilbd', 'bview'

Description: organic percentage of each soil layer

Variable name: **phi**

Common blocks: 'spec.bthree'

Subroutines: 'bci', 'bcone', 'bmenu', 'bread', 'bview'

Description: internal angle of friction

Variable name: **phib**

Common blocks: 'spec.bthree'

Subroutines: 'bci', 'bcone', 'bmenu', 'bread', 'bview'

Description: remoulded angle of friction

Variable name: **ppt**(140)

Common blocks: 'spec.btwo'

Subroutines: 'binfil', 'bmenu', 'bread', 'bview'

Description: hourly precipitation values

Variable name: **press**

Common blocks: 'spec.btwo'

Subroutines: 'bevapor', 'bmenu', 'bread', 'bview'

Description: barometric pressure

Variable name: **rci**

Common blocks: 'spec.bout'

Subroutines: 'bci', 'binfil'

Description: cone index passed to 'binfil' where it is modified to rating cone index

Variable name: **ri**

Subroutines: 'bevapor'

Description: Richardson index

Variable name: **s1**

Common blocks: 'spec.bout'

Subroutines: 'bci', 'bcone'

Description: intermediate calculation passed from 'bcone'

Variable name: **s2**

Common blocks: 'spec.bout'

Subroutines: 'bci', 'bcone'

Description: intermediate calculation passed from 'bcone'

Variable name:**s3**

Subroutines: 'bci'

Description: intermediate calculation

Variable name: **satcon**(11)

Common blocks: 'spec.bgrnd'

Subroutines: 'binfil', 'bcntl', 'bhydcon'

Description: saturated hydraulic conductivity for each layer

Variable name: **saz**

Subroutines: 'bevapor'

Description: solar azimuth in radians

Variable name: **sbp**(11)

Common blocks: 'spec.bnow', 'spec.bd1'

Subroutines: 'bschars'

Description: standard deviation for distribution of bubbling pressure (air entry matric potential) for soils 1..11

Variable name: **scp**(11)

Common blocks: 'spec.bnow', 'spec.bd1'

Subroutines: 'bschars'

Description: standard deviation of clay content percentage distribution for soil types 1..11

Variable name: **sicf**

Subroutines: 'bevapor'

Description: insolation adjustment due to zenith angle, surface slope and surface aspect angle

Variable name: **slope**

Common blocks: 'spec.btwo'

Subroutines: 'balter', 'bevapor', 'bmenu', 'bread', 'bview'

Description: slope angle (degrees)

Variable name: **sm**(11)

Common blocks: 'spec.bnow', 'spec.bd1'

Subroutines: bschars'

Description: standard deviation of mineral bulk density distribution of soil types 1..11

Variable name: **sp**(11)

Common blocks: 'spec.bgrnd'

Subroutines: 'bschars', 'bsoilbd'

Description: sand content percentage for each soil layer

Variable name: **speed**

Subroutines: 'bevapor'

Description: wind speed (cm/sec)

Variable name: **sr**(j)

Common blocks: 'spec.bgrnd'

Subroutines: 'bcntl', 'bhydcon', 'binfil', 'bschars'

Description: saturated water content calculated in 'bschars'

Variable name: **ssatcon**(11)

Common blocks: 'spec.bnow", 'spec.bdata'

Subroutines: 'bschars'

Description: standard deviation of saturated hydraulic conductivity distribution

Variable name: **ssp**(11)

Common blocks: 'spec.bnow', 'spec.bd1'

Subroutines: 'bschars'

Description: standard devaition of sand content percentage distribution of soil types 1..11

Variable name: **ssr**(11)

Common blocks: 'spec.bnow', 'spec.bd1'

Subroutines: 'bschars'

Description: standard deviation of distribution for the saturation for soils 1..11

Variable name: **sum**

Subroutine: 'bci'

Description: intermediate calculation

Variable name: **sun**

Subroutines: 'bevapor'

Description: calculated insolation value

Variable name: **surfacaz**

Common blocks: 'spec.btwo'

Subroutines: 'balter', 'bevapor', bmenu', 'bread', 'bview'

Description: surface azimuth angle

Variable name: **t**

Subroutines: 'bevapor'

Description: time

Variable name: **ta**

Subroutines: 'bevapor'

Description: air temperature in kelvin

Variable name: **tac**

Subroutines: 'bevapor'

Description: air temperature in celsius

Variable name: **tak**

Subroutines: 'bevapor'

Description: air temperature in kelvin

Variable name: **tcom**(20)

Common blocks: 'spec.btwo'

Subroutines: 'binfil', 'bmenu', 'bread', 'bview'

Description: thickness of soil cells

Variable name: **tf**(10,20)

Common blocks: 'spec.btwo'

Subroutines: 'binfil', 'bmenu', 'bread', 'bview'

Description: initail soil moisture content of each cell

Variable name: **timer**

Subroutines: 'bevapor'

Description: sun's hour angle in radians

Variable name: **tops**

Subroutines: 'bhydcon'

Description: higher point on 20 point O-k curve that observation is next to

Variable name: **tyme**

Subroutines: 'bevapor'

Description: time in hours

Variable name: **ua**

Common blocks: 'spec.bthree'

Subroutines: 'bci', 'bcone', 'bread'

Description: air entry matric potential

Variable name: **uw**

Common blocks: 'spec.bout'

Subroutines: 'bci', 'binfil'

Description: soil water potential

Variable name: **water**

Subroutines: 'bevapor'

Description: amount of precipital water (mm) calculated for use in solving insolation

Variable name: **wet**

Subroutines: 'bevapor'

Description: moisture content of surface layer

Variable name: **x**(j,i)

Common blocks: 'spec.bman'

Subroutines: 'bcntl', 'binfil'

Description: O values for 20 point O-U curve passed to 'binfil'

Variable name: **xii**

Subroutines: 'bhydcon'

Description: counter on the 20 point soil - moisture characteristic curve

Variable name: **xl**

Subroutines: 'bevapor'

Description: latent heat of evaporation as function of air and ground temperature

Variable name: **xnew**(j,i)

Common blocks: 'spec.bman'

Subroutines: 'bcntl', 'bsmcurv'

Description: θ values for 20 point θ-ψ curve passed from 'bsmcurv'

Variable name: **xxx**(10,6)

Common blocks: 'spec.btwo'

Subroutines: 'bevapor', 'bmenu', 'bread', 'bview'

Description: times of met. observations (alg. 20211d)

Variable name: **y**(j,i)

Common blocks: 'spec.bdat2'

Subroutines: 'bcntl'

Description: specified matric potentials for use in θ-ψ curve

Variable name: **yjj**

Subroutines: 'bhydcon'

Description: counter on the 20 point soil - moisture characteristic curve

Variable name: **yyy**(10,6)

Common blocks: 'spec.btwo'

Subroutines: 'balter', 'bevapor', 'bmenu', 'bread', 'bview'

Description: met. observations (alg. 20211d)

Variable name: **z**(11,20)

Common blocks: 'spec.bfour'

Subroutines: 'bhydcon', 'binfil'

Description: hydraulic conductivity

Variable name: **za**

Common blocks: 'spec.btwo'

Subroutines: 'bevapor', 'bmenu', 'bread', 'bview'

Description: site zenith angle

Variable name: **zim**

Subroutines: 'bevapor'

Description: solar zenith angle

Variable name: **zz**

Common blocks: 'spec.bthree'

Subroutines: 'bci', 'bmenu', 'bread', 'bview'

Description: depth of cone tip

A2 : VSAS2

The following section contains a list of the parameters and variables used in the VSAS2. It is arranged in the alphabetical order of the variable name. Each name is accompanied by the common block they may be in, the subroutines they are used in and a brief-description of their function.

Variable name: **aal**

Common blocks: 'spec.vcone'

Subroutines: 'vcone', 'vread',

Description: cone length =1.48in

Variable name: **adepth**

Common blocks: 'spec.vcom2'

Subroutines: 'vblkvol', 'vmenu', 'vread', 'vview'

Description: depth of top layer (alg. 20221a)

Variable name: **alpha**(5)

Subroutines: 'vblkvol'

Description:

Variable name: **alpha**

Common blocks: 'spec.vcone'

Subroutines: 'vcone', 'vread'

Description: cone tip half angle =15^0

Variable name: **answ**

Subroutines: 'vcntl', 'vmenu'

Description: character variable set to 'y' or 'n' by the user in reply to a system query

Variable name: **area**(16)

Common blocks: 'spec.vgeom'

Subroutines: 'vblkvol'

Description: area of each increment

Variable name: **asd1**

Subroutines: 'vblkvol'

Description: set to half the height increase above the centre of an element

Variable name: **asd2**

Subroutines: 'vblkvol'

Description: set to half the height increase below the centre of an element

Variable name: **aslen**(5,16)

Common blocks: 'spec.cgeom'

Subroutines: 'vblkvol'

Description: average distance between the mid-points of two side by side segments

Variable name: **avdep**(5,16,5)

Common blocks: 'spec.vgeom'

Subroutines: 'vblkvol', 'vdrain', 'vinit', 'vouta'

Description: average depth of the mid point of each element

Variable name: **avdrop**(5,16,5)

Common blocks: 'spec.vgeom'

Subroutines: 'vblkvol', 'vinit'

Description: average height drop across an element

Variable name: **avele**(5,16,5)

Common blocks:'spec.vgeom'

Subroutines: 'vblkvol', 'vdtain', 'vinit', 'vlatflo', 'vlatflo2', 'vouta'

Description: average elevation of each increment up the slope

Variable name: **avlen**(5,16)

Common blocks: 'spec.vgeom'

Subroutines: 'vblkvol', 'vinit'

Description: average length of segment

Variable name: **avvol**(5,16,5)

Common blocks: 'spec.vgeom'

Subroutines: 'vblkvol', 'vdrain', 'vinit', 'vlatflo2', 'vouta'

Description: average volume of each element

Variable name: **avwid**(5,16)

Common blocks: 'spec.vgeom'

Subroutines: 'vblkvol', 'vinit', 'vouta'

Description: average width of each increment

Variable name: **c**

Common blocks: 'spec.vcone'

Subroutines: 'vcone'

Description: soil cohesion

Variable name: **clevel**

Common blocks: 'spec.vgeom'

Subroutines: 'vdrain', 'vinit', 'vmenu', 'vread', 'vview'

Description: channel level (alg 20221a)

Variable name: **depmax**(5,16)

Common blocks: 'spec.vcom2'

Subroutines: 'vblkvol', 'vmenu', 'vread', 'vview'

Description: depth of each increment (alg 20221a)

Variable name: **diff**(5)

Subroutines: 'vblkvol'

Description:

Variable name: **e1**

Common blocks: 'spec.vcone2'

Subroutines: 'vci', 'vcone'

Description: intermediate calculation passed from 'vcone'

Variable name: **e2**

Common blocks: 'spec.vcone2'

Subroutines: 'vci', 'vcone'

Description: intermediate calculation passed from 'vcone'

Variable name: **e3**

Subroutines: 'vci'

Description: intermediate calculation

Variable name: **e4**

Common blocks: 'spec.vcone2'

Subroutines: 'vci', 'vcone'

Description: intermediate calculation passed from 'vcone'

Variable name: **elev**(5,0:20,5)

Subroutines: 'vblkvol'

Description: local array set to the mid elevation between 2 elements

Variable name: **fac**

Subroutines: 'vdrain', 'vlatflo', 'vlatflo2'

Description: conversion factor cm/hr to m/hr

Variable name: **fexist**

Subroutines: 'vcntl', 'vmenu'

Description: logical test variable used to test if a file exists

Variable name: **flow**(5,6)

Common blocks: 'spec.vwater'

Subroutines: 'vcntl', 'vdrain'

Description: outflow from each layer

Variable name: **fname**

Subroutines: 'vcntl', 'vmenu'

Description: character variable representing a file name

Variable name: **gg**

Common blocks: 'spec.vcone'

Subroutines: 'vcone','vmenu', 'vread', 'vview'

Description: soil shear modulus

Variable name: **hlen**(5)

Common blocks: 'spec.vindex'

Subroutines: 'vblkvol', 'vmenu', 'vread', 'vview'

Description: mean length of segment

Variable name: **horiz**(5,6)

Common blocks: 'spec.vcom1'

Subroutines: 'vblkvol', 'vinit', 'vmenu', 'vread', 'vview'

Description: length of segment side (alg. 20221a)

Variable name: **iend**

Common blocks:

Subroutines: 'vcntl'

Description: number of hours for duration of run

Variable name: **imonth**

Common blocks: 'spec.vtime'

Subroutines: 'vcntl', 'vread', 'vview', 'vouta'

Description: month run is to simulate

Variable name: **isgno**

Common blocks: 'spec.vindex'

Subroutines: 'vcntl', 'vmenu', 'vread', 'vview', 'vouta', 'voutb'

Description: segment number

Variable name: **it**

Common blocks: 'spec.vindex'

Subroutines: 'vcntl'

Description: iteration number

Variable name: **ix**

Subroutines: 'vcntl', 'vdrain', 'vouta'

Description: counter set to 1 for vertical flow and 2 for slopewise flow

Variable name: **ixm**

Common blocks: 'spec.vindex'

Subroutines: 'vcntl', 'vouta'

Description:

Variable name: **jno**

Common blocks: 'spec.vindex'

Subroutines: 'vblkvol', 'vdrain', 'vinit', 'vlatflo', 'vlatflo2', 'vmenu', 'vread', 'vview'

Description: number of soil layers

Variable name: **jx**

Subroutines: 'vdrain', 'vblkvol'

Description: counter

Variable name: **k**

Subroutines: 'vblkvol'

Description: counter

Variable name: **k1**

Subroutines: 'vcntl', 'vmenu'

Description: sets input - output status

Variable name: **km**

Subroutines: 'vblkvol'

Description: counter

Variable name: **kno**

Common blocks: 'spec.vindex'

Subroutines: 'vblkvol', 'vcntl', vdrain', 'vinit', 'vmenu', 'vouta'
'vread', 'vview'

Description: total number of segments

Variable name: **krep**

Common blocks: 'spec.vindex'

Subroutines: 'vcntl', 'vdrain', 'vlatflo', 'vlatflo2', 'vmenu', 'vread', 'vview'

Description: number of coarse iterations in one hour

Variable name: **ksum**

Common blocks:

Subroutines: 'vblkvol'

Description:

Variable name: **lrep**

Common blocks: 'spec.vindex'

Subroutines: 'vcntl', 'vdrain', 'vlatflo', 'vlatflo2', 'vmenu', 'vread', 'vview'

Description: number of finer iterations in each 'krep' iteration

Variable name: **moi**

Subroutines: 'vcntl'

Description: character variable set to a file name

Variable name: **ni**

Common blocks: 'spec.vindex'

Subroutines: 'vblkvol', 'vdrain', 'vlatflo', 'vlatflo2', vouta'

Description: counter used to descrbe the increments up the slope

Variable name: **nl**

Common blocks: 'spec.vindex'

Subroutines: 'vblkvol'

Description: counter

Variable name: **nlimk**(k)

Common blocks: 'spec.vkgr'

Subroutines: 'vblkvol', 'vinit', 'vlatflo', 'vlatflo2', 'vmenu', 'vouta', 'vread', 'vview'

Description: number of increments up the slope of segment 'k'

Variable name: **nn**

Subroutines: 'vblkvol', 'vmenu', 'vread', 'vview'

Description: counter

Variable name: **nnn**

Subroutines: 'vcntl'

Description: counter

Variable name: **nouta**

Common blocks: 'spec.vindex'

Subroutines: 'vcntl', 'vmenu', 'vread', 'vview'

Description: option for printout of segment geometry, if nouta = 0 print, otherwise do not

Variable name: **ns**

Subroutines: 'vdrain'

Description: counter

Variable name: **ntmp**

Common blocks: 'spec.vindex'

Subroutines: 'vcntl', 'vmenu', 'vread', 'vview'

Description: option for printout of hourly segment status, if ntmp = 0 print, otherwise do not

Variable name: **nx**

Subroutines: 'vdrain'

Description: counter

Variable name: **phi**

Common blocks: 'spec.vcone'

Subroutinés: 'vcone', 'vmenu', 'vread'

Description: internal angle of soil's friction

Variable name: **phib**

Common blocks: 'spec.vcone'

Subroutines: 'vci', 'vcone', 'vmenu', 'vread', 'vview'

Description: remoulded internal angle of soil's friction

Variable name: **pnet**

Common blocks: 'spec.vwater'

Subroutines: 'vcntl', 'vinter', 'vouta', 'vsurflo'

Description: net hourly precipitation values

Variable name: **pnr**

Common blocks: 'spec.vwater'

Subroutines: 'vcntl', 'vdrain'

Description: amount of water available to infiltrate the permeable surface area of the segment

Variable name: **poros**(5,16,5)

Common blocks: 'spec.vphys'

Subroutines: 'vcon', 'vdrain', 'vinit', 'vmenu', 'vouta', 'vread', 'vview', 'vxmatrx'

Description: porosity of each element

Variable name: **pos**(3)

Common blocks: 'spec.vdata'

Subroutines: 'vinit', 'vmenu', 'vread', 'vview'

Description: position up the slope that hydrological parameters change

Variable name: **potdif**

Subroutines: 'vdrain', 'vlatflo2'

Description: potential difference between elements

Variable name: **poten**(5,16,5)

Common blocks: 'spec.vphys'

Subroutines: 'vdrain', 'vlatflo', 'vlatflo2', 'vouta', 'vxmatrx'

Description: soil water potential for each element

Variable name: **precip**(1500)

Common blocks: 'spec.vwater'

Subroutines: 'vcntl', 'vinter', 'vmenu', 'vread', 'vview'

Description: hourly precipitation values

Variable name: **pres**(5,16,5)

Common blocks: 'spec.vphys'

Subroutines: 'vdrain'

Description: positive pressure of each soil element

Variable name: **q**

Common blocks:'spec.vwater'

Subroutines: 'vcntl'

Description: segment outflow rate for each hourly iteration (ft^3/hr)

Variable name: **qq**

Subroutines: 'vcntl'

Description: cumulative segment outflow rate for each hourly iteration (m^3/hr)

Variable name: **qqq**(40)

Subroutines: 'vcntl'

Description: cumulative outflow from each segment

Variable name: **qqqq**

Subroutines: 'vcntl'

Description: total outflow for cumulated segments for simulated period (m^3)

Variable name: **rain**

Subroutines: 'vcntl', 'vinter', 'voutb'

Description: precipitation value calculated after interception in 'vinter'

Variable name: **s1**

Common blocks: 'spec.vcon2'

Subroutines: 'vci', 'vcone'

Description: intermediate calculation passed from 'vcone'

Variable name: **s2**

Common blocks: 'spec.vcone2'

Subroutines: 'vci', 'vcone'

Description: intermediate calculation passed from 'vcone'

Variable name: **s3**

Subroutines: 'vci'

Description: intermediate calculation

Variable name: **sgarea**

Common blocks: 'spec.vcom2'

Subroutines: 'vblkvol', 'vcntl', 'vmenu', 'vouta', 'vread', 'vview'

Description: segment area

Variable name: **shift**(5)

Common blocks: 'spec.vkgr'

Subroutines: 'vdrain', 'vblkvol'

Description:

Variable name: **smcmax**(5,16,5)

Common blocks: 'spec.vphys'

Subroutines: 'vdrain', 'vinit'

Description: maximum soil water content for each element

Variable name: **smcvol**(5,16,5)

Common blocks: 'spec.vphys'

Subroutines: 'vdrain', 'vinit', 'vlatflo', 'vlatflo2', 'vouta'

Description: actual water content of each element

Variable name: **stonec**(5)

Common blocks: 'spec.vgeom'

Subroutines: 'vblkvol', 'vinit', 'vmenu', 'vread', 'vview'

Description: stone content of each layer

Variable name: **sum**

Subroutines: 'vci'

Description: intermediate calculation

Variable name: **sumar**(5,16)

Common blocks: 'spec.vgeom'

Subroutines: 'vblkvol', 'vdrain'

Description: area of each element

Variable name: **tpbv**

Subroutines: 'vcon', 'vdrain'

Description: soil moisture content passed from 'vdrain' to 'vcon' to calculate the hydraulic conductivity

Variable name: **ua**

Common blocks: 'spec.vcone'

Subroutines: 'vci', 'vcone', 'vread'

Description: air entry matric potential

Variable name: **uw**

Subroutines: 'vci', 'vouta'

Description: soil water potential passed from 'vouta' as an argument

Variable name: **vci**

Description: cone index

Variable name: **xflo**(5,16,5)

Subroutines: 'vdrain', 'vlatflo', 'vlatflo2'

Description: slopewise flow from each element

Variable name: **xi**(5,16)

Common blocks: 'spec.com1'

Subroutines: 'vblkvol', 'vlatflo2', 'vmenu', 'vread', 'vview'

Description: distance from stream of each increment (alg. 20221a)

Variable name: **xwidth**(5,6)

Common blocks: 'spec.vcom2'

Subroutines: 'vblkvol', 'vmenu', 'vread', 'vview'

Description: widths of segment (alg 20221a)

Variable name: **yi**(5,16)

Common blocks: 'spec.vcom2'

Subroutines: 'vblkvol', 'vmenu', 'vread', 'vview'

Description: elevation of each increment (alg 20221)

Variable name: **zz**

Common blocks:'spec.vcone'

Subroutines: 'vcone', 'vmenu', 'vread'

Description: depth of interest (i.e. cone tip)

A3 Code description for optimum route management

A3.1 Subroutine 'rroute' (algorithm 2042a)

This subroutine controls the optimum route management scheme and is called by 'rcntl'

Variable name: **ans1**

Common blocks: 'rgrid'

Subroutines: 'rroute', 'rread'

Description: character variable set in 'rread' by the system asking if the user wishes to return to the previous levels menu (y or n)

A3.2 Subroutine 'rread2' (algoritm 2042b)

This subroutine asks the user to be certain that a full set of data files exists . The user is asked if they would like to return to the previous level's menu. If the user continues, the subroutine asks for the maximum number of files to be accessed and the time the RCI value is to be read at. The data files are then accessed (data.1,....data.max, where max = maximum number of files defined by the user.

Variable name: **ans1**

Common blocks: 'rgrid'

Subroutines: 'rroute', 'rread'

Description: character variable set in 'rread' by the system asking if the user wishes to return to the previous levels menu (y or n)

Variable name: **t**

Description: time RCI is required at

Variable name: **n**

Description: maximum number of files to be accessed

Variable name: **name**

Description: character variable set to the name of the data file to be accessed

Variable name: **try(I)**

Common blocks: 'spec.tset'

Subroutines: 'rread'

Description: character variable array containing the logical sequence of 'data.n' files

Variable name: **kk**

Description: counter used to read the initial information from the 'data.n' files until the required time 't' is reached

Variable name: **lab**

Description: dummy variable used in reading initial unwanted information from the 'data.n' files

Variable name: **ir(n)**

Common blocks: 'rrci'

Subroutines: 'rread', 'rrcid'

Description: RCI value read from data file 'data.n'

A3.3 Subroutine 'rrcid' (algorithms 2042c-i)

This subroutine calculates an optimum route from a specified grid and library of data files. It asks the user to specify start and finish squares.

Variable name: **fname**

Description: character variable set by the user to the name of the grid

Variable name: **fexist**

Description: a logical test variable - testing if sa file exists

Variable name: **liml**

Description: lower limit of RCI that vehicle can move on

Variable name: **jj**

Description: counter for number of columns

Variable name: **ii**

Description: counter for number of rows

Variable name: **igg(15,15)**

Common blocks: 'spec.rgrid'

Subroutines: 'rrcid'

Description: igg(posc,posr) is the RCI at the time 't' of the grid square located at column 'posc' and row 'posr'

Variable name: **ir(30)**

Common blocks: 'spec.rrci'

Subroutines: 'rread', rrcid'

Description: RCI read in 'rread' for each identifying integer

Variable name: **ig(15,15)**

Common blocks: 'spec.rgrid'

Subroutines: 'rread', 'rcreate', 'rrcid', 'rsave'

Description: identifying integer (relating to 'data.n') for each grid square

Variable name: **check(15,15)**

Description: each grid square is allocated a check number (move number kkk) when considered as a potential move

Variable name: **j**

Description: counter

Variable name: **i**

Description. counter

Variable name: **k1**

Description: sets input - output status

Variable name: **cs**

Description: column start, i.e. start square is identified by col. cs, row rs

Variable name: **rs**

Description: row start, i.e. start square is identified by col. cs, row rs

Variable name: **cf**

Description: column finish, i.e finish square is identified by col. cf, row rf

Variable name: **rf**

Description: row finish

Variable name: **posc**

Description: present column position, i.e. present grid square id identified by col. posc, row posr

Variable name: **posr**

Description: present row position

Variable name:**kkk**

Description: counter for number of moves - set to a max. of 30 at present

Variable name: **rtc(30)**

Description: rtc(kkk) is column position of move 'kkk'

Variable name: **rtr(30)**

Description: rtr(kkk) is row position of move 'kkk'

Variable name: **trackc(30)**

Common blocks: 'spec.rtrav'

Subroutines: 'rrcid', 'rprint'

Description: trackc(kkk) is the column position of move 'kkk'

Variable name: **trackr(30)**

Common blocks: 'spec.rtrav'

Subroutines: 'rrcid', 'rprint'

Description: trackr(kkk) is the row position of move 'kkk'

Variable name: **biasc**

Description: bias of present grid position column from finish column

Variable name: **biasr**

Description: bias of present grid position row from finish row

Variable name: **bc**

Description: variable set according to column bias for which direction to look for next move and bias the move in a direction towards the finish square

Variable name: **br**

Description: variable set according to row bias for which direction to look for next move and bias the move in a direction towards the finish square

Variable name: **iac**

Description: set to ensure that bias in column direction is an integer

Variable name: **iar**

Description: set to ensure that bias in row direction is an integer

Variable name: **iac**

Description: set to ensure that bias in column direction is an integer

Variable name: **total**

Description: sum of total bias in both rows and columns

Variable name: **diff**

Description: calculates the dominant direction of movement, i.e.
north <-> south or west <-> east

Variable name: **movec**

Description: next move is set to column 'movec'

Variable name: **mover**

Description: next move is set to row 'mover'

Variable name: **kan**

Description: recalculation of the biased RCI value in the grid square

Variable name: **kbn**

Description: recalculation of the biased RCI value in the grid square

Variable name: **kcn**

Description: recalculation of the biased RCI value in the grid square

Variable name: **ian**

Description: recalculation of the biased RCI value in the grid square
(north <-> south movt. in algorithm 2042g)

Variable name: **ibn**

Description: recalculation of the biased RCI value in the grid square
(north <-> south movt. in algorithm 2042g)

Variable name: **icn**

Description: recalculation of the biased RCI value in the grid square
(north <-> south movt. in algorithm 2042g)

Variable name: **jan**

Description: recalculation of the biased RCI value in the grid square
(west <-> east movt. in algorithm 2042g)

Variable name: **jbn**

Description: recalculation of the biased RCI value in the grid square
(west <-> east movt. in algorithm 2042g)

Variable name: **jcn**

Description: recalculation of the biased RCI value in the grid square
(west <-> east movt. in algorithm 2042g)

A3.4 Subroutine 'rprint' (algorithm 2042j)

This subroutine prints out the optimum route track in terms of grid square moves. It is called by 'rrcid'.

Variable name: **j**

Description: counter for number of moves

Variable name: **trackc(30)**

Common blocks: 'spec.rtrav'

Subroutines: 'rrcid', 'rprint'

Description: trackc(kkk) is the column position of move 'kkk'

Variable name: **trackr(30)**

Common blocks: 'spec.rtrav'

Subroutines: 'rrcid', 'rprint'

Description: trackr(kkk) is the row position of move 'kkk'

A4 Code description

A4.1 Subroutine 'rcreate' (algorithms 2031a &2031b)

This subroutine creates a grid to the users' specifications and is called by 'rcntl'.

Variable name: **kit**

Common blocks:'spec.rgrid'

Subroutines: 'rcreate', 'rread'

Description: check variable used in checking whether a file already exists

Variable name: **k1**

Description: sets input - output status

Variable name: **ii**

Common blocks: 'spec.rgrid'

Subroutines: 'rcreate', 'rread', 'rsave', 'rcheck'

Description: number of columns in grid

Variable name: **jj**

Common blocks: 'spec.rgrid'

Subroutines: 'rcreate', 'rread', 'rsave', 'rcheck'

Description: number of rows in grid

Variable name: **i**

Description: counter

Variable name: **j**

Description: counter

Variable name: **ig (15,15)**

Common blocks: 'spec.rgrid'

Subroutines: 'rread', 'rcreate', 'rrcid', 'rsave'

Description: identifying integer (relating to 'data.n') for square ig(i,j)

A4.2 Subroutine 'rcheck' (algorithms 2032b & 2032c)

This subroutine allows the use to edit a grid file and is called by 'rcntl'.

Variable name: **k10, k11, k12, k13**

Description: sets input - output status

Variable name: **ii**

Common blocks: 'spec.rgrid'

Subroutines: 'rcreate', 'rread', 'rsave', 'rcheck'

Description: number of columns in grid

Variable name: **jj**

Common blocks: 'spec.rgrid'

Subroutines: 'rcreate', 'rread', 'rsave', 'rcheck'

Description: number of rows in grid

Variable name: **i**

Description: counter

Variable name: **j**

Description: counter

Variable name: **ig (15,15)**

Common blocks: 'spec.rgrid'

Subroutines: 'rread', 'rcreate', 'rrcid', 'rsave', 'rcheck'

Description: identifying integer (relating to 'data.n') for square ig(i,j)

A4.3 Subroutine 'rsave' (algorithms 2031c &2032d)

This subroutine saves the grid specifications and identifiers in a file specified by the user. It is called by 'rcntl'.

Variable name: **ans**

Common blocks: 'spec.rgrid'

Subroutines: 'rsave'

Description: character check variable set as either 'y' or 'n' in response to a query from the system

Variable name: **name**

Common blocks: 'spec.rgrid'

Subroutines: 'rsave', 'rrcid'

Description: name of output file

Variable name: **fexist**

Description: a logical test variable - testing if a file exists

Variable name: **ii**

Common blocks: 'spec.rgrid'

Subroutines: 'rcreate', 'rread', 'rsave', 'rcheck'

Description: number of columns in grid

Variable name: **jj**

Common blocks: 'spec.rgrid'

Subroutines: 'rcreate', 'rread', 'rsave', 'rcheck'

Description: number of rows in grid

Variable name: **i**

Description: counter

Variable name: **j**

Description: counter

Variable name: **ig (15,15)**

Common blocks: 'spec.rgrid'

Subroutines: 'rread', 'rcreate', 'rrcid', 'rsave', 'rcheck'

Description: identifying integer (relating to 'data.n') for square ig(i,j)

A4.4 Subroutine 'rread' (algorithm 2032a)

This subroutine reads an existing grid and is called by 'rcntl'.

Variable name: **name**

Common blocks: 'spec.rgrid'

Subroutines: 'rsave', 'rrcid'

Description: name of output file

Variable name: **fexist**

Description: a logical test variable - testing if a file exists

Variable name:**ii**

Common blocks: 'spec.rgrid'

Subroutines: 'rcreate', 'rread', 'rsave', 'rcheck'

Description: number of columns in grid

Variable name: **jj**

Common blocks: 'spec.rgrid'

Subroutines: 'rcreate', 'rread', 'rsave', 'rcheck'

Description: number of rows in grid

Variable name: **i**

Description: counter

Variable name: **j**

Description: counter

Variable name: **ig (15,15)**

Common blocks: 'spec.rgrid'

Subroutines: 'rread', 'rcreate', 'rrcid', 'rsave', 'rcheck'

Description: identifying integer (relating to 'data.n') for square ig(i,j)

Variable name: **kit**

Common blocks:'spec.rgrid'

Subroutines: 'rcreate', 'rread'

Description: check variable used in checking whether a file already exists

A5 Code description

A5.1 Subroutine 'tcntl' (algorithms 2041b, 2041c & 2041d)

This subroutine controls the evaluation of RCI over a specified route such as that shown in **figure 6.17a.** The subroutine accesses RCI data files created by the BSSS or the VSAS2 and outputs the route evaluation to an output file of the users choice. Subroutine 'tcntl' is called by subroutine 'link'.

Variable name: **name**
Description: charcter variable set to the name of the output file
specified by the user from a system prompt

Variable name: **answ1**
Subroutines: 'tcntl', 'bmsg'
Description: character variable passed from 'bmsg2', used as a system check that the user really means to overwrite an existing file

Variable name: **fexist**
Description: a logical test variable - testing if a file exists

Variable name: **n**
Common blocks: 'tinput'
Subroutines: 'tcntl', 'tred'
Description: number of nodes on specified route

Variable name: **i**
Description: counter

Variable name: **pos(i)**

Common blocks: 'tinput'

Subroutines: 'tcntl', 'tred'

Description: distance along a route of node number 'i'. It is read in 'tred' and passed into 'tcntl'

Variable name: **id(i)**

Common blocks: 'tinput'

Subroutines: 'tcntl', 'tred'

Description: identifying integer used to access 'data.1', 'data.2',..'data.i', where 'i' is the node number. It is read in 'tred' and passed into 'tcntl'

Variable name: **pos(i)**

Common blocks: 'tinput'

Subroutines: 'tcntl', 'tred'

Description: distance along a route of node number 'i'. It is read in 'tred' and passed into 'tcntl'

Variable name: **id(i)**

Common blocks: 'tinput'

Subroutines: 'tcntl', 'tred'

Description: identifying integer used to access 'data.1', 'data.2',..'data.i', where 'i' is the node number. It is read in 'tred' and passed into 'tcntl'

Variable name:**try(i)**

Common blocks: 'tset'

Subroutines: 'tcntl'

Description: character variable array containing the logical sequence of 'data.n' files

Variable name: **it**

Description: counter used to read the initial information from the 'data.n' files until the required time 't(i)' is reached

Variable name: **a**

Description: dummy variable used in reading initial unwanted information from the 'data.n' files

Variable name: **rci(i)**

Common blocks: 'tinput'

Subroutines: 'tcntl'

Description: rating cone index value of node 'i'

A5.2 Subroutine 'tred' (algorithm 2041a)

This subroutine asks the user to specify a route for the route evaluation scheme in 'tred1', and in 'tred2' an existing saved route is read for use in 'tcntl'. The number of nodes is asked for and for each node: distance along the route, identifying interger and time.

Variable name: **name**

Description: charcter variable set to the name of the output file
specified by the user from a system prompt

Variable name: **answ1**

Subroutines: 'tcntl', 'bmsg'

Description: character variable passed from 'bmsg2', used as a system check that the user really means to overwrite an existing file

Variable name: **fexist**

Description: a logical test variable - testing if a file exists

Variable name: **n**

Common blocks: 'tinput'

Subroutines: 'tcntl', 'tred'

Description: number of nodes on specified route

Variable name: **i**

Description: counter

Variable name: **pos(i)**

Common blocks: 'tinput'

Subroutines: 'tcntl', 'tred'

Description: distance along a route of node number 'i'. It is read in 'tred' and passed into 'tcntl'

Variable name: **id(i)**

Common blocks: 'tinput'

Subroutines: 'tcntl', 'tred'

Description: identifying integer used to access 'data.1', 'data.2',..'data.i', where 'i' is the node number. It is read in 'tred' and passed into 'tcntl'

Variable name: **pos(i)**

Common blocks: 'tinput'

Subroutines: 'tcntl', 'tred'

Description: distance along a route of node number 'i'. It is read in 'tred' and passed into 'tcntl'

Variable name: **id(i)**

Common blocks: 'tinput'

Subroutines: 'tcntl', 'tred'

Description: identifying integer used to access 'data.1', 'data.2',..'data.i', where 'i' is the node number. It is read in 'tred' and passed into 'tcntl'

APPENDIX B : Listing of the Bristol Off-Road Trafficability System

The following FORTRAN code is listed in alphabetical order of subroutine names. The following system has been used to identify subroutines with greater ease:-

BSSS - all start with 'b'
VSAS2 - all start with 'v'
Optimum route and grid - all start with 'r'
Route Evaluation - all start with 't'

```
c
c*********************************************************
c
c SUBROUTINE    BALTER
c
c*********************************************************
c
      subroutine balter
c
c
c This subroutine adjusts those varibles in 'bevapor' which are input to
c the BRISTOL SOIL STRENGTH SCHEME in inappropriate but user friendly
c units:-e.g. temperatures, slope angles and surface azimuths.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c June, 1987.
c
c
c 'balter' is called by 'bcntl'
c
      include 'spec.bfour'
      include 'spec.btwo'
      write(14,*)' alter'
      do 116 j=1,max(1)
  116 yyy(j,1)=yyy(j,1)+273.15
      do 117 j=1,max(2)
  117 yyy(j,2)=yyy(j,2)*0.01
      do 118 j=1,max(4)
  118 yyy(j,4)=yyy(j,4)+273.15
      do 119 j=1,max(6)
  119 yyy(j,6)=yyy(j,6)*100.0
      slope=slope*3.14159/180.0
      surfacaz=surfacaz*3.14159/180.0
       i=1
       goto 9935
 9936  if(i.gt.6) goto 9934
 9935  imax=max(i)
       j=1
       goto 9932
 9933  if(j.eq.imax) goto 9931
 9932  fmm(j,i)=(yyy(j+1,i)-yyy(j,i))/(xxx(j+1,i)-XXX(J,I))
       bbb(j,i)=yyy(j,i)-fmm(j,i)*xxx(j,i)
       j=j+1
       goto 9933
 9931  i=i+1
       if(i.eq.5) i=6
       goto 9936
 9934  CONTINUE
      return
      end
```

```
c
c******************************************************
c
c SUBROUTINE    BCI
c
c******************************************************
c
      subroutine bci
c
c
c This subroutine calculates the variable part of the RCI equation
(Cochrane)
c in the BRISTOL SOIL STRENGTH SCHEME.
c Created by J.E. Cochrane
c at the Geography Department, Bristol University, England.
c April, 1987.
c
c
c 'bci' is called by 'binfil'
c
      include 'spec.bout'
      include 'spec.bthree'
c
c
      e3=(ua+uw)*tan(phib)
      s3=(e1-e2+e3)*e4
      sum=(s1*s2)-s3
      ci=sum
      rci=sum
c
c
      return
      end
```

```
c
c*******************************************************
c
c SUBROUTINE    BCNTL
c
c*******************************************************
c
      subroutine bcntl
c
c
c This subroutine controls the BRISTOL SOIL STRENGTH SCHEME.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c September, 1987.
c
c
c 'bcntl' is called by 'link'
c
c 'bcntl' calls 'bmenu' for editing input files
c 'bcntl' calls 'bread1', 'bread2', 'bschars', 'bsoilbd', 'bsmcurv',
c                'bhydcon', 'balter', & 'binfil'
c
       character ans*1
       character miss(30)*10
       logical fexist
       include 'spec.bchar'
      include 'spec.bgrnd'
      include 'spec.bman'
      include 'spec.bone'
      include 'spec.bdat2'
      include 'spec.bset'
c
      open(9,file='smcurv')
      il=1
      ival=1
      k=2
c
c 'bmenu' offers the user the choice of either editing an input file or
c running the scheme.
c
      call bmenu
      if(ii.eq.4)goto 100
      iset=0
    2 write(6,*)'WHICH DATA FILE IS TO RUN?'
      read(5,*)miss(il)
      inquire(file=miss(il),name=miss(il),exist=fexist)
      if(.not.fexist)call bmsg1
      if(.not.fexist)goto 2
    5 write(6,*)'OUTPUT FOR THIS RUN WILL BE IN data.',ival
      write(6,*)'ALL OUTPUT FILES ARE data.n, WHERE n MAY BE CHANGED'
      bfile(il)=nset(ival)
      write(6,*)'DO YOU WISH TO CHANGE n? y or n'
      read(5,*,err=5)ans
      if(ans.ne.'y')goto 3
    4   write(6,*)'INPUT VALUE FOR n 1-30'
        k1=1
        read(5,*,iostat=k1,err=4)ival
        bfile(il)=nset(ival)
    3 write(6,*)'DO YOU WANT TO MAKE ANOTHER RUN? y or n'
      il=il+1
      ival=ival+1
      k1=1
      write(14,*)'bfile',bfile(il)
      read(5,*,iostat=k1,err=3)ans
      if(ans.eq.'y')goto 2
      icount=il-1
      do 90 il=1,icount
      open(2,file=miss(il))
      rewind 2
      if(iset.ne.1)call bread1
c
c To facilitate the stochastic variation of the soil parameters
c the nag routine g0ccf is used to randomly initialise the other nag
routines
c
      call g05ccf
      do 10 j=1,nos
c
c 'bschars' searches an existing data bank for the mean and s.d.
c of those soil parameters to be calculated by stochastic variation.
c
      call bschars
      write(14,*)'sr',sr(j)
c
c 'bsoilbd' calculates the soil bulk density and the basic 10 point
c suction moisture curve.
c
      call bsoilbd
      write(14,*)'soilbd'
      do 20 i=1,9
         ax(j,i)=aax(j,i)
         write(14,*)ax(j,i)
         y(j,i)=y(1,i)
   20 CONTINUE
c
c
c set the end points of the suction moisture curve
c
         ax(j,10)=sr(j)
         y(j,10)=bp(j)*10.18
c
c
c 'bsmcurv' calculates a 20 point suction moisture curve at equal moisture
c intervals.
c
      call bsmcurv
      write(14,*)'smcurv'
c
      do 30 i=1,20
         x(j,i)=xnew(j,i)
         y(j,i)=ynew(j,i)
         write(9,*)i,x(j,i),y(j,i),j
   30 CONTINUE
c
c
c 'bhydcon' calculates the soil moisture characteristic curve O-K.
```

```
c
      call bhydcon
c     call bprint1
   10 CONTINUE
c
      if(iset.eq.0)call bread2
c
      idays=days
c
c 'balter' adjusts those variables in 'bevapor' which are input in
c user friendly units.
c
      call balter
c
c 'binfil' calculates the movement of water throuth the soil profile.
c 'binfil' also controls the evaporation calculation and RCI.
c
      call binfil
      close(2)
   90 CONTINUE
c
      return
c
c
  100 stop
c
c
      end
```

```
c
c****************************************
c
c    CONE SUBROUTINE
c
c*****************************************
      subroutine bcone
c
c
c This subroutine calculates the parts of the RCI equation (Cochrane)
c which are dependent on only the soil parameters in the BRISTOL SOIL
c STRENGTH SCHEME.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c April, 1987.
c
c
c 'bcone' is called by 'binfil'
c
      include 'spec.bthree'
      include 'spec.bout'
c
c
      real m
       alpha=(22*2*alpha)/(7*360)
      phi=(22*2*phi)/(7*360)
      phib=(22*2*phib)/(7*360)
      m=(4*sin(phi))/(3*(1+sin(phi)))
      b1=(tan(alpha)+tan(phi))*tan(alpha)*8
      a1=1-m
      a2=2-m
      a3=3-m
      b2=d*d*a1*a2*a3
      b3=(gg*(1/tan(phi)))**m
      b4=(gamma)**a1
      s1=(b1*b3*b4)/b2
c
      d1=((c/gamma)*(1/tan(phi))+aal+zz)**a3
      d2=((c/gamma)*(1/tan(phi))+zz)**a3
      d3=((c/gamma)*(1/tan(phi))+zz)**a2
      d3=d3*a3*aal
      s2=d1-d2-d3
c
      e1=c*tan(alpha)*(1/tan(phi))
      e2=ua*tan(phi)
      e4=(4*aal*aal*tan(alpha))/(d*d)
c
  876 CONTINUE
      return
      end
```

```
c
c*****************************************************
c
c SUBROUTINE    BEVAPOR
c
c*****************************************************
c
      subroutine bevapor
c
c
c This subroutine calculates the evaporation of a surface from a
c non-isothermal equation (Khale,A.B.) developed for use by the
c U.S. Corps of Engineers (Balick,M. et al.). It is used in the
c BRISTOL SOIL STRENGTH SCHEME.
c
c Original by Balick, M. et al.
c Adapted by J.E. Cochrane
c November, 1986.
c
c
c 'bevapor' is called by 'binfil'
c
      include 'spec.bsix'
      include 'spec.btwo'
      include 'spec.bout'
      include 'spec.bfour'
c
c
      dimension clr(8)
c     esat(ta)=6.108*exp(ac*(TA-273.15)/(TA-bc))
C    to INITIALIZE-VAriableS-and-CONSTANTS
        data clr/0.04,0.08,0.17,0.20,0.22,0.24,0.24,0.25/
        data  acl/82.2,87.1,52.5,39.0,34.7,23.8,11.2,15.4/
        data  bcl/.079,.148,.112,.063,.104,.159,-.167,.028/
        data sigma,pi,ac,bc/8.12e-11,3.141593,
     &      17.269,35.86/
        data cc/0.261/
        data last,grav,ksq,cp/24,980.0,0.16,0.24/
c
       day=days+ifix(tyme/24.0)
       tyme=tyme-(day-days)*24.0
       bb=-2.4E-4
       ieof=0
       ibug=0
       all=0.040
       facth=(1000./press)**0.286
       t0=2.0*pi*(day-1.0)/365.0
       decl=0.006918-0.399912*cos(t0)+0.070257*sin(t0)
     &-0.006758*cos(2.0*t0)+0.000907*sin(2.0*t0)-0.002697*cos(3.0*t0)
     &+0.001480*sin(3.0*t0)
       elf=(lat/180*pi)
       timer=(tyme/12*pi)+pi
       if(timer.gt.2.*pi)timer=timer-2.*pi
       aa=cos(decl)*cos(elf)*cos(timer)
       bb=sin(decl)*sin(elf)
       c=aa+bb
       zim=acos(c)
C-------------------------------------------------------------
```

```
C     TO SOLVE-SOLAR-AZIMUTH
C
        xnum=-cos(decl)*sin(timer)
        xdnom=cos(elf)*sin(decl)-sin(elf)*cos(timer)
        saz=atan(xnum/xdnom)
        if(.not.(xnum.lt.0.0.and.xdnom.gt.0.0)) goto  9944
        saz=saz+pi
        goto  9945
 9944 if(.not.(xnum.gt.0.0.and.xdnom.gt.0.0)) goto  9943
        saz=saz-pi
 9943 CONTINUE
 9945 CONTINUE
C-------------------------------------------------------------
C     to CALCULATE-SLOPE-ATMOS-ATTEM-AND-CLOUD-ADJUSTMENTS
C
        sicf=cos(zim)*cos(slope)+sin(zim)*sin(slope)
     &  *cos(saz-surfacaz)
        if(.not.(sicf.lt.0.0.or.cos(zim).le.0.0)) goto  9941
        sun=0.0
        goto  9942
 9941 m=1/cos(zim)
        if(.not.(m.ge.0.0)) goto  9939
        tal=0.02023
        if(day.ge.92.0 .and. day.le.152.0)tal=-0.02290
      xn=amod(tyme,24.)
      ntabl=1
      assign 30 to i9930
      goto  9930
 30     ta=yn
      ntabl=2
      assign 40 to i9930
      goto  9930
  40    rh=yn
        td=282.7
        water=exp(0.07074*(td-273.15)+tal)
        ab=0.271*(water*m)**0.303
        a0=0.085-0.247*alog10(press/1000.*1./m)
        arg1=(1.-alb)*cos(zim)*(1.-ab)*0.349+(1.-a0)*(1.- al1)/
     & (1.-a0*0.2)*0.651*cos(zim)
        goto  9940
 9939 arg1=1.0
 9940 qp=2.0*arg1
        qo=qp*sicf
        if(.not.(ncloud.eq.0)) goto  9937
        sun=qo
        goto  9938
 9937 xn=amod(tyme,24.0)
      ntabl=3
      assign 10 to i9930
      goto  9930
 10     cloud=yn
        arg2=-(bcl(ncloud)-.059)*m
        ctf=(acl(ncloud)/94.4)*exp(arg2)
        sun=qo-((cloud*cloud)*(qo-qo*ctf))
 9938 CONTINUE
 9942 CONTINUE
      goto  9879
 9930 CONTINUE
```

```
C    TO GET-TABLE-VALUES
C
       imax=max(ntabl)
       ij=1
       if(.not.(xn.ge.xxx(imax,ntabl))) goto  9928
       yn=yyy(imax,ntabl)
       goto  9929
 9928 if(ij.eq.imax+1) goto  9927
       jj=ij
       if(.not.(xxx(ij,ntabl).lt.xn)) goto  9925
       ij=ij+1
       goto  9926
 9925 if(.not.(xxx(ij,ntabl).eq.xn)) goto  9924
       yn=yyy(jj,ntabl)
       ij=imax+1
       goto  9926
 9924 if(.not.(xxx(ij,ntabl).gt.xn)) goto  9923
       jj=jj-1
       yn=fmm(jj,ntabl)*xn+bbb(jj,ntabl)
       ij=imax+1
 9923 CONTINUE
 9926 goto  9928
 9927 CONTINUE
 9929 goto i9930
C-------------------------------------------------------------
 9879 CONTINUE
C    to ATMOSPHERIC-INFRARED-EMISSION-ATERM
      t=tyme
       xn=amod(t,24.0)
       ntabl=2
C
C      GET-TABLE-VALUES
       assign 9867 to i9930
       goto 9930
C
 9867 rh=yn
      xn=amod(t,24.0)
      ntabl=1
C
C      GET-TABLE-VALUES
       assign  9866 to i9930
       goto 9930
C
 9866 ta=yn
       xn=amod(t,24.0)
       ntabl=3
C
C      GET-TABLE-VALUES
       assign  9865 to i9930
       goto  9930
C
 9865 cloud=yn
       tak=ta
       tac=(tak-273.15)
       ea=6.108*rh*exp((ac*tac)/(tak-bc))
       alphi=(0.61+0.05*sqrt(ea))*(1.0+(clr(ncloud)*(cloud**2)))
       downir=0.8132e-10*tak**4*alphi
C-------------------------------------------------------------
```

```
C    to CALCULATE-CONVECTION-HTERM
      xn=amod(t,24.0)
      ntabl=4
      assign 20 to i9930
      goto  9930
 20   teml=yn
C     if(.not.(teml.gt.ta)) goto  9863
C
      xn=amod(t,24.0)
      ntabl=6
C
C     GET-TABLE-VALUES
      assign  9862 to i9930
      goto  9930
C
 9862 speed=yn
      tak=ta
      zash=za
      tsk=teml
      rhoa=-0.001*0.348*press/tak
      thetaz=tak*facth
      thetas=tsk*facth
      dtheta=(thetaz-thetas)/zash
      du=speed/zash
      thetav=(thetaz+thetas)/2.0
      ri=grav*dtheta/(thetav*du**2)
      coe1=15.0
      coe2=1.175
      ex=.75
      if(tsk.gt.tak)goto 31
      if(ri.gt.0.2)ri=.19999
      coe1=5.0
      coe2=1.0
      ex=2.0
   31 hter=rhoa*ksq*zash**2*du
     &*(coe2*(1.0-coe1*ri)**ex)
      hterm=hter*cp*dtheta
C-------------------------------------------------------------
C    to CALCULATE-EVAPORATIVE-HEAT-LOSS-DTERM
C
      if(.not.(teml.gt.ta)) goto  9860
      xn=amod(t,24.0)
      ntabl=2
C
C     GET-TABLE-VALUES
      assign  9859 to i9930
      goto  9930
C
 9859 rh=yn
      xn=amod(t;24.0)
      ntabl=1
C
C     GET-TABLE-VALUES
      assign  9858 to i9930
      goto  9930
C
 9858 ctema=yn
      ktempa=ctema
      ctema=ctema-273.15
      ktempg=teml
      es=exp((ac*(ktempg-273.15))/(ktempg-bc))*6.1071
      ea=exp((ac*ctema)/(ktempa-bc))*6.1071*rh
      dg=0.622/press*(ea-es)*wet/za
      xl=597.3-0.566*(ctema+ktempg-273.15)/2.0
      dterm=hter*xl*dg
      evap=((sun+downir)-hterm)/xl
     evap=evap/6000.
 898 CONTINUE
      return
9860 evap=0.
      return
      end
```

```
c
c***********************************************************
c
c SUBROUTINE    BHYDCON
c
c***********************************************************
c
      subroutine bhydcon
c
c
C This subroutine calculates hydraulic conductivity for each layer
C from the given soil moisture characteristic curve.
C Uses the Millington and Quirk method

c Original by S. Howes
c Adapted by J.E. Cochrane
c at Geography Department, University of Bristol, England.
c November, 1986.
c
c
c 'bhydcon' is called by 'bcntl'
c
      include 'spec.bgrnd'
      include 'spec.bfour'
      include 'spec.bman'
      do 845 i=1,20
      iij=20-i+1
      xii=x(j,iij)
      tops=0.
      bots=0.
         do 846 k=1,20
         jf=20-k+1
         if(yjj.le.0)yjj=1
 846     bots=((2*k-1)*yjj**(-2))+bots
      ii
          o 847 k=ii,20
         jf=20-k+1
         if(yjj.le.0)yjj=1
  847    tops=((2*k+1-2*i)*yjj**(-2))+tops
      jt=20-i+1
  845 z(j,jt)=satcon(j)*(x(j,ii)/
     & sr(j))*tops/bots
      return
      end
```

```
c
c********************************************************
c
c SUBROUTINE   BINFIL
c
c********************************************************
c
      subroutine binfil
c
c
c This subroutine calculates the flow of water through a soil profile.
c Values of suction are passed out to calculate the RCI in the BRISTOL
c SOIL STRENGTH SCHEME.
c Uses Richard's Equation.
c Calls a non-isothermal evaporation algorithm.
c Original M.G. Anderson
c Adapted by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c December, 1986.
c
c
c 'binfil' is called by 'bcntl'
c
c 'binfil' calls 'bread3'
c 'binfil' calls 'bcone'
c 'binfil' calls 'bevapor'
c 'binfil' calls 'bci'
c
      character nam*10
      include 'spec.bchar'
      include 'spec.btwo'
      include 'spec.bthree'
      include 'spec.bgrnd'
      include 'spec.bman'
      include 'spec.bfive'
      include 'spec.bout'
       include 'spec.bfour'
      include 'spec.bset'
c
c
c
      write(14,*)'il',il
      write(14,*)'bfile',bfile(il)
      nam=bfile(il)
      write(14,*)'nset',nset(il),il
      open(13,file=nset(il))
       open(9,file=nam,err=900)
c
       write(14,*)'infil'
       if(iset.eq.0)call bread3
       pqr=999
       nl1=nl+1
      rain1=0.0
      nq=20
      nq=nq-1
      nmi=1
      do 99 iz=1,nmi
      do 823 i=1,nl
      tf(iz,i)=(tf(iz,i)*sr(iz))/100
      theta(i)=tf(iz,I)
  823 write(14,*)'theta',theta(i),i
      cumdrn=0.
      cinfil=0.
      sumd=0.
      icount =0
      br=amr-alr
      sog(1)=theta(1)/sr(1)
      su=theta(1)
      colin=exp(4.605+((2.123+.008*cp(1)-.693*alog(su*72.59))/
     & (.149+.002*cp(1))))
 1051 FORMAT( ' COLLiNS rci =  ',F10.4)
      tr=exp(al+bl*alog(theta(1)*72.59))
 1112 FORMAT(F4.2,F6.4,F14.12,F6.4,F14.12)
      dist(1)=tcom(1)/2
      rtot=0.0
      pptt=0.0
      anfilt=0.
      tg=0.0
      depth(1)=dist(1)
      do 5 i=2,nl
      anflux(i)=0.0
c
c *** CALCULATES TOTAL DEPTH (depth) TO MID POINT OF CELL
c
      depth(i)=depth(I-1)+0.5*(tcom(I-1)+tcom(i))
c
c *** CALCULATES DISTANCE (dist) BETWEEN MIDPOINTS OF EACH CELL
c
    5 dist(i)=0.5*(tcom(I-1)+tcom(i))
      itmax=immax*3600/af
      wati=0.
      do 478 i=1,nl
c
c *** CALCULATES TOTAL COLUMN WATER CONTENT
c
  478 wati=tcom(i)*theta(i)+wati
      evapi=0.
      do 261 i=1,nq
c
c *** CALCULATES GRADIENT OF SUCTION - MOISTURE CURVE (g)
c     AND SOIL - MOISTURE CHARACTERISTIC CURVE FOR EACH LAYER
c
      g(i)=(y(1,i+1)-y(1,i))/(x(1,i+1)-x(1,i))
  261 gz(i)=(z(1,i+1)-z(1,i))/(x(1,i+1)-x(1,i))
      do 262 i=1,nq
      g3(i)=(y(3,i+1)-y(3,i))/(x(3,i+1)-x(3,i))
      gz3(i)=(z(3,i+1)-z(3,i))/(x(3,i+1)-x(3,i))
      g2(i)=(y(2,i+1)-y(2,i))/(x(2,i+1)-x(2,i))
  262 gz2(i)=(z(2,i+1)-z(2,i))/(x(2,i+1)-x(2,i))
      phib=25.0
      call bcone
      open(11,file='swr')
      write(11,*)'swr opened'
      write(11,*)y(1,20)
      uw=(y(1,20)*100)/(2.54)
      call bci
      rci=ci*0.7
      write(14,*)y(1,20),rci
      uw=0.0
      call bci
      rci=ci*0.7
      write(14,*)’min rci’,rci
      write(11,*)’phi’,phi
      do 777 ja=1,3
      do 776 ia=1,20
      uw=(y(ja,ia)*100)/(2.54)
      call bci
      write(11,*)x(ja,ia),ia,ja,ci,y(ja,ia)
  776 CONTINUE
  777 CONTINUE
      write(14,*)’itmax’,itmax,’immax’,immax
      do 410 ii=1,itmax
      do 10 i=1,nl
c
c *** CALCULATES VOLUMETRIC WATER CONTENT (vol) OF EACH CELL
c
   10 vol(i)=tcom(i)*theta(I)
      icount=icount+af
      tg=tg+af
      t=ii
      nlaa=nla1+1
      nlab=(nla1+nla2+1)
c
c *** THE NEXT SECTION CALCULATES SOIL WATER POTENTIAL (sw)
c     I.E. SUCTION, HYDRAULIC POTENTIAL (hpot) AND
c     CONDUCTIVITY (cond) FOR EACH OF THE SOIL WATER CONTENTS
c     (theta) FOR EACH CELL AT EACH TIME ITERATION
c
       do 15 i=1,nla1
      do 16 j=1,nq
      if(theta(i).ge.x(1,j).and.theta(i).lt.x(1,j+1))
     &swp(i)=y(1,j)+g(j)*(theta(i)-x(1,j))
   16 CONTINUE
      hpot(i)=swp(i)-depth(i)
      do 17 j=1,nq
      if(theta(i).gt.x(1,j).and.theta(i).le.x(1,j+1))
     & cond(i)=z(1,j)+gz(j)*(theta(i)-x(1,j))
   17 CONTINUE
   15 CONTINUE
      do 515 i=nlaa,nl
      do 516 j=1,nq
      if(theta(i).ge.x(2,j).and.theta(i).lt.x(2,j+1)) swp(i)=
     & y(2,j)+g2(j)*(theta(i)-x(2,j))
  516 CONTINUE
      hpot(i)=swp(i)-depth(i)
      do 517 j=1,nq
      if(theta(i).gt.x(2,j).and.theta(i).le.x(2,j+1)) cond(i)=
     & z(2,j)+gz2(j)*(theta(i)-x(2,j))
  517 CONTINUE
  515 CONTINUE
      do 615 i=nlab,nl
      do 616 j=1,nq
      if(theta(i).ge.x(3,j).and.theta(i).lt.x(3,j+1)) swp(i)=y(3,j)
     & +g3(j)*(theta(i)-x(3,j))
  616 CONTINUE
      hpot(i)=swp(i)-depth(i)
      do 617 j=1,nq
      if(theta(i).gt.x(3,j).and.theta(i).le.x(3,j+1)) cond(i)=
     & z(3,j)+gz3(j)*(theta(i)-x(3,j))
  617 CONTINUE
  615 CONTINUE
      uw=(swp(1)*100/(2.54))
      call bci
      rci1=ci*0.7
      uw=(swp(2)*100)/(2.54)
      call bci
      rci=ci*0.7
      c1 = exp(4.605+((2.123+.008*cp(1)-.693*alog(theta(1)*72.59))/
     & (.149+.002*cp(1))))
      c2 = exp(4.605+((2.123+.008*cp(1)-.693*alog(theta(3)*72.59))/
     & (.149+.002*cp(1))))
 1110 FORMAT(' SOIL TYPE = ',I2,' SATCON  = ',F14.12)
 1111 FORMAT(F7.3,F7.4,F7.4,F14.9,2F10.4,F7.4,2F10.4)
      t1=t*af/3600.
      rain=0.0
      aq=amr-alr
      iaq=aq
      ar=1
      do 910 i=1,iaq
      if(ppt(i).eq.0.0)go to 910
      j=amr-alr-i
      ai=i
      if(ai.ne.aq.and.ppt(i).eq.0.0)go to 910
      ajt=alr+i-1
      ajr=amr-j
      if(t1.ge.ajt.and.t1.lt.ajr)rain=ppt(i)/3600.0
  910 CONTINUE
      pptt=pptt+rain*af
c
c *** CALCULATES THE AVERAGE CONDUCTIVITY (avcond) BETWEEN CELLS
c
      do 210 i=2,nl
  210 avcond(i)=(cond(i-1)*tcom(i-1)+cond(i)*tcom(i))
     &/(tcom(i-1)+tcom(i))
      flux(nl1)=cond(nl)
c     hpot(0)=(hpot(1)/3*2)
c
c *** THE NEXT SECTION CALCULATES THE FLOW (flux) OF WATER
c     BETWEEN CELLS
c
      do 220 i=1,nl
      if(dist(i).eq.0.0) dist(i)=0.1
  220 flux(i)=(hpot(i-1)-hpot(i))*avcond(i)/dist(i)
c
c *** CALCULATES THE INFILTRATION RATE (bncap)
c
      bncap=(0.0-hpot(1))*0.5*(satcon(1)+cond(1))/dist(1)
      sumd=(rain-anfilt)*af+sumd
      rain2=rain
      if(rain1.ne.rain2)sumd=0.0
      rain1=rain2
      if(sumd.lt.0.0)detain=0.0
      detain=sumd
      if(rain.gt.0.0) go to 350
      runoff=0.0
      sog(1)=theta(1)/sr(1)
      tyme=af*t/3600.0
      if(detain.le.0.0) go to 330
      call bevapor
      anfilt=bncap
      flux(1)=anfilt
      go to 390
  330 anfilt=0.0
      call bevapor
c     if(swp(1).le.(0-10))evap=0.0
c
c *** SETS THE FLUX FROM THE TOP OF THE COLUMN TO THE
c     EVAPORATION CALCULATED IN bevapor.f
c
      flux(1)=evap*(-1.0)
      go to 390
  350 evap= 0.0
  818 FORMAT("evap  = ",F13.11," At TiME ",F7.3," rain = ",F7.5)
      anfilt=bncap
      if(rain.lt.bncap.and.detain.le.0.0)anfilt=rain
      flux(1)=anfilt
      if(detain.lt.detcap) go to 390
      sumd=detcap
      runoff=0.0
c
c *** CALCULATES THE RUNOFF (runoff) IN EXCESS OF THE
c     INFILTRATION CAPACITY (bncap)
c
      if(rain.gt.bncap)runoff=(rain-bncap)*af
      rtot=rtot+runoff
  390 CONTINUE
      do 320 i=1,nl
      anflux(i)=flux(i)-flux(i+1)
      anflux(i)=anflux(i)*af
c
c *** THE NEXT SECTION UPDATES THE WATER CONTENT OF EACH CELL
c     AND THEN CHECKS EACH CELL TO ENSURE THAT THERE IS NOT
c     MORE THAN 100% SATURATION
c
       theta(i)=(vol(i)+anflux(i))/tcom(i)
      if(theta(i).gt.sr(1).and.i.le.nla1)theta(i)=sr(1)
      if(theta(i).gt.sr(2).and.i.gt.nla1.and.i.le.nlab)theta(i)=sr(2)
      if(theta(i).gt.sr(3).and.i.gt.nlab)theta(i)=sr(3)
  320 CONTINUE
c
c *** THE NEXT SECTION CUMULATES THE DRAINAGE AND EVAPORATION
c     INTO HOURLY CHECK TOTALS
c
      cumdrn=cumdrn+flux(nl1)*af
      evapi=evap*af+evapi
      cinfil=cinfil+anfilt*af
      if(icount.ne.3600) go to 410
      icount=0
      t=t*af/3600
c     write(09,1111)t,sog(1),theta(1),swp(1),rci1,c1,theta(2),rci,c2
```

```
    write(9,*)rcil,evapi,cumdrn,cinfil
434 FORMAT(///)
834 FORMAT(' TRAFFICABILITY PREDICTIONS ****'/' SURFACE STRENGTH
   & =',F10.3/' RATING CONE INDEX =',F10.3,' REL. SAT =',F10.6/)
    if(tg.eq.86400.0)tg=0.0
    nlac=nlab-1
    do 815 i=1,nlal
815 sog(i)=theta(i)/sr(1)
    do 816 i=nlaa,nlac
816 sog(i)=theta(i)/sr(2)
    do 817 i=nlab,nl
817 sog(i)=theta(i)/sr(3)
    write(14,*)rcil
    write(14,*)rcil
503 FORMAT(' RUNOFF TOTAL =',F10.7,' IN THE LAST HOUR'///)
    rtot=0.0
    watn=0.
    do 479 i=1,nl
479 watn=tcom(i)*theta(i)+watn
    bal=watn-wati-cinfil+evapi+cumdrn
    bal=(bal*100.)/watn
232 FORMAT(' BALANCE CHECK AS % COL. WATER VOL. =',F16.10,' %'/)
481 FORMAT(' BALANCE CHECK ON PROFILE WATER STATUS =',F12.7)
611 FORMAT(' CUMULATIVE PRECIP = ',F8.4)
410 CONTINUE
415 FORMAT(i6,6F14.9)
500 FORMAT('    **** TIME FROM START =',F7.3,'HOURS')
610 FORMAT('EVAPORATION=',F13.11)
502 FORMAT('   LAYER ** DEPTH   **   SOIL WATER ** MOISTURE   **  HYD CO
   &ND   **  NET FLUX  **  REL SAT  **')
 60 FORMAT(' CUM.INFIL =',F10.6,' CUM. DRAIN. =',F10.6////)
    if(t.ge.10) go to 897
 99 CONTINUE
898 CONTINUE
897 close(9)
900 write(14,*)'at 900 in binfil.f'
    return
    end
```

```
c
c*********************************************************
c
c SUBROUTINE     BMENU
c
c*********************************************************
c
       subroutine bmenu
c
c
c This subroutine allows the user to create and edit input files
c for the BRISTOL SOIL STRENGTH SCHEME
c Created by J.E. Cochrane
c Date October 1987
c at Geography Department, Bristol University, England.
c
c 'bmenu' is called by 'bcntl'
c 'bmenu' calls 'bview' and 'bread'
c
      character name*20
      character answ*1
      logical fexist
      include 'spec.bone'
      include 'spec.btwo'
      include 'spec.bthree'
      include 'spec.bgrnd'
c
    1 write(6,*)'MENU SELECTION'
      write(6,*)'1. CREATE A NEW DATA FILE'
      write(6,*)'2. CHANGE AN EXISTING DATA FILE'
      write(6,*)'3. RUN BRISTOL SSS'
      write(6,*)'4. EXIT WITHOUT A RUN'
      k1=1
      read(5,*,iostat=k1,err=1)ii
      goto(5,430,1000,1010)ii
      goto(5,430,1000,1000)ii
    5 k1=1
   10 write(6,*)'INPUT NUMBER OF LAYERS'
      write(6,*)'PRESENT VALUE IS ',nos
      read(5,*,iostat=k1,err=10)nos
      if(jr.gt.o)goto 435
   11 write(6,*)'INPUT TOPOGRAPHIC A/S VALUE e.g. FLAT - 1.0'
      write(6,*)'SLOPE - 1.2, VALLEY BOTTOM - 0.8'
      write(6,*)'PRESENT VALUE IS ',aos
      k1=1
      read(5,*,iostat=k1,err=11)aos
      if(jr.gt.0)goto 435
   15 do 25 i=1,nos
   20 write(6,*)'INPUT SOIL TYPE FOR LAYER',i
      write(6,*)'PRESENT VALUE IS ',isty(i)
      k1=1
   25 read(5,*,iostat=k1,err=20)isty(i)
      if(jr.gt.0)goto 435
   27 do 35 i=1,nos
   30 write(6,*)'INPUT ORGANIC MATTER % FOR LAYER ',i
      write(6,*)'PRESENT VALUE IS ',op(i)
      k1=1
   35 read(5,*,iostat=k1,err=30)op(i)
```

```
      if(jr.gt.0)goto 435
   40 write(6,*)'INPUT SLOPE ANGLE IN DEGREES '
      write(6,*)'PRESENT VALUE IS ',slope
      k1 =1
      read(5,*,iostat=k1,err=40)slope
      if(jr.gt.0)goto 435
   45 write(6,*)'INPUT ZENITH ANGLE '
      write(6,*)'PRESENT VALUE IS ',za
      k1=1
      read(5,*,iostat=k1,err=45)za
      if(jr.gt.0)goto 435
   50 write(6,*)'INPUT CLOUD TYPE 1..8 '
      write(6,*)'PRESENT VALUE IS ',ncloud
      k1=1
      read(5,*,iostat=k1,err=50)ncloud
      if(jr.gt.0)goto 435
   55 write(6,*)'INPUT BAROMETRIC PRESSURE (MILLIBARS) '
      write(6,*)'PRESENT VALUE IS ',press
      k1=1
      read(5,*,iostat=k1,err=55)press
      if(jr.gt.0)goto 435
   60 write(6,*)'INPUT SURFACE AZIMUTH (DEGREES) '
      write(6,*)'PRESENT VALUE IS ',surfacaz
      k1=1
      read(5,*,iostat=k1,err=60)surfacaz
      if(jr.gt.0)goto 435
   65 write(6,*)'INPUT LATITUDE OF SITE (DEGREES) '
      write(6,*)'PRESENT VALUE IS ',lat
      k1=1
      read(5,*,iostat=k1,err=65)lat
      if(jr.gt.0)goto 435
   70 write(6,*)'INPUT JULIAN DAY OF RUN '
      write(6,*)'PRESENT VALUE IS ',days
      k1=1
      read(5,*,iostat=k1,err=70)days
      if(jr.gt.0)goto 435
   71 write(6,*)'INPUT ALBEDO OF SURFACE '
      write(6,*)'PRESENT VALUE IS ',alb
      k1=1
      read(5,*,iostat=k1,err=71)alb
      if(jr.gt.0)goto 435
c

c
c
  120 write(6,*)'INPUT NUMBER OF AIR TEMPERATURE OBSERVATIONS '
      write(6,*)'PRESENT VALUE IS ',max(1)
      k1=1
      read(5,*,iostat=k1,err=120)max(1)
      if(jr.gt.0)goto 435
  125 write(6,*)'INPUT NUMBER OF RELATIVE HUMIDITY OBSERVATIONS '
      write(6,*)'PRESENT VALUE IS ',max(2)
      k1=1
      read(5,*,iostat=k1,err=125)max(2)
      if(jr.gt.0)goto 435
  130 write(6,*)'INPUT NUMBER OF CLOUD COVER OBSERVATIONS '
      vrite(6,*)'PRESENT VALUE IS ',max(3)
      k1=1
      read(5,*,iostat=k1,err=130)max(3)
      if(jr.gt.0)goto 435
  140 write(6,*)'INPUT NUMBER OF GROUND TEMP OBSERVATIONS'
      write(6,*)'PRESENT VALUE IS ',max(4)
      k1=1
      read(5,*,iostat=k1,err=140)max(4)
      if(jr.gt.0)goto 435
  145 write(6,*)'INPUT NUMBER OF WIND SPEED OBSERVATIONS '
      write(6,*)'PRESENT VALUE IS ',max(6)
      k1=1
      read(5,*,iostat=k1,err=145)max(6)
      if(jr.gt.0)goto 435
c
  198 do 200 i=1,max(1)
  202 write(6,*)'INPUT TIME OF AIR TEMP OBSERVATION NO. ',i
      write(6,*)'PRESENT VALUE IS ',xxx(i,1)
      k1=1
      read(5,*,iostat=k1,err=202)xxx(i,1)
  210 write(6,*)'INPUT AIR TEMPERATURE FOR OBSERVATION NO. ',i
      write(6,*)'PRESENT VALUE IS ',yyy(i,1)
      k1=1
  200 read(5,*,iostat=k1,err=210)yyy(i,1)
      if(jr.gt.0)goto 435
  215 do 230 i=1,max(2)
  222 write(6,*)'INPUT TIME OF RELATIVE OBSERVATION NO. ',i
      write(6,*)'PRESENT VALUE IS ',xxx(i,2)
      k1=1
      read(5,*,iostat=k1,err=222)xxx(i,2)
  225 write(6,*)'INPUT RELATIVE HUMIDITY OBSERVATION NO. ',i
      write(6,*)'PRESENT VALUE IS ',yyy(i,2)
      k1=1
  230 read(5,*,iostat=k1,err=225)yyy(i,2)
      if(jr.gt.0)goto 435
  231 do 240 i=1,max(3)
  232 write(6,*)'INPUT TIME OF CLOUD COVER OBSERVATION NO. ',i
      write(6,*)'PRESENT VALUE IS ',xxx(i,3)
      k1=1
      read(5,*,iostat=k1,err=232)xxx(i,3)
  235 write(6,*)'INPUT VALUE OF CLOUD COVER OBSERVATION NO. ',i
      write(6,*)'PRESENT VALUE IS ',yyy(i,3)
      k1=1
  240 read(5,*,iostat=k1,err=235)yyy(i,3)
      if(jr.gt.0)goto 435
  241 do 250 i=1,max(4)
  242 write(6,*)'INPUT TIME OF GROUND TEMP OBSERVATION NO. ',i
      write(6,*)'PRESENT VALUE IS ',xxx(i,4)
      k1=1
      read(5,*,iostat=k1,err=242)xxx(i,4)
  245 write(6,*)'INPUT GROUND TEMP OBSERVATION NO. ',i
      write(6,*)'PRESENT VALUE IS ',yyy(i,4)
      k1=1
  250 read(5,*,iostat=k1,err=245)yyy(i,4)
      if(jr.gt.0)goto 435
  251 do 260 i=1,max(6)
  252 write(6,*)'INPUT TIME OF WIND SPEED OBSERVATION NO. ',i
      write(6,*)'PRESENT VALUE IS ',xxx(i,6)
      k1=1
      read(5,*,iostat=k1,err=252)xxx(i,6)
  255 write(6,*)'INPUT WIND SPEED OBSERVATION NO. ',i
      write(6,*)'PRESENT VALUE IS ',yyy(i,6)
      k1=1
  260 read(5,*,iostat=k1,err=255)yyy(i,6)
      if(jr.gt.0)goto 435
c
  265 write(6,*)'INPUT NUMBER OF CELLS IN THE VERTICAL COLUMN '
      write(6,*)'PRESENT VALUE IS ',nl
      k1=1
      read(5,*,iostat=k1,err=265)nl
      do 268 i=1,nos
  267 write(6,*)'INPUT NUMBER OF CELLS IN LAYER ',i
      write(6,*)'PRESENT VALUE IS ',nla(i)
      k1=1
  268 read(5,*,iostat=k1,err=267)nla(i)
      if(jr.gt.0)goto 435
  270 write(6,*)'INPUT SURFACE DETENTION CAPACITY (m) '
      write(6,*)'PRESENT VALUE IS ',detcap
      k1=1
      read(5,*,iostat=k1,err=270)detcap
      if(jr.gt.0)goto 435
  275 write(6,*)'INPUT ITERATION TIME INTERVAL (secs) '
      write(6,*)'PRESENT VALUE IS ',af
      k1=1
      read(5,*,iostat=k1,err=275)af
      if(jr.gt.0)goto 435
c
  276 do 288 i=1,nl
  277 write(6,*)'INPUT INITIAL SOIL MOISTURE AS % FOR CELL',i
      write(6,*)'PRESENT VALUE IS ',tf(1,i)
      k1=1
  288 read(5,*,iostat=k1,err=277)tf(1,i)
      if(jr.gt.0)goto 435
  289 do 280 i=1,nl
  278 write(6,*)'INPUT THICKNESS (m) OF CELL NO. ',i
      write(6,*)'PRESENT VALUE IS ',tcom(i)
      k1=1
  280 read(5,*,iostat=k1,err=278)tcom(i)
      if(jr.gt.0)goto 435
  400 write(6,*)'INPUT INTERNAL ANGLE OF FRICTION PHI'
      write(6,*)'PRESENT VALUE IS ',phi
      k1=1
      read(5,*,iostat=k1,err=400)phi
      write(6,*)'INPUT REMOULDED INTERNAL ANGLE OF FRICTION  PHIB'
      write(6,*)'PRESENT VALUE IS ',phib
      k1=1
      read(5,*,iostat=k1,err=400)phib
      if(jr.gt.0)goto 435
  405 write(6,*)'INPUT SOIL COHESION'
      write(6,*)'PRESENT VALUE IS ',c
      k1=1
      read(5,*,iostat=k1,err=405)c
      if(jr.gt.0)goto 435
  410 write(6,*)'INPUT SOIL SHEAR MODULUS'
      write(6,*)'PRESENT VALUE IS ',gg
      k1=1
      read(5,*,iostat=k1,err=410)gg
      if(jr.gt.0)goto 435
  415 write(6,*)'INPUT SOIL DENSITY'
      write(6,*)'PRESENT VALUE IS ',d
      k1=1
      read(5,*,iostat=k1,err=415)d
      if(jr.gt.0)goto 435
  420 write(6,*)'INPUT DEPTH OF CONE TIP'
      write(6,*)'PRESENT VALUE IS ',zz
      k1=1
      read(5,*,iostat=k1,err=420)zz
      if(jr.gt.0)goto 435
c
  285 write(6,*)'INPUT NUMBER OF HOURS OF RAIN '
      write(6,*)'PRESENT VALUE IS ',ir
      k1=1
      read(5,*,iostat=k1,err=285)ir
      if(jr.gt.0)goto 435
  286 do 290 i=1,ir
  287 write(6,*)'INPUT PRECIPITATION (m) FOR HOUR ',i
      write(6,*)'PRESENT VALUE IS ',ppt(i)
      k1=1
  290 read(5,*,iostat=k1,err=287)ppt(i)
      if(jr.gt.0)goto 435
  295 write(6,*)'INPUT RAIN START TIME '
      write(6,*)'PRESENT VALUE IS ',alr
      k1=1
      read(5,*,iostat=k1,err=295)alr
      if(jr.gt.0)goto 435
  300 write(6,*)'INPUT RAIN STOP TIME '
      write(6,*)'PRESENT VALUE IS ',amr
      k1=1
      read(5,*,iostat=k1,err=300)amr
      if(jr.gt.0)goto 435
  305 write(6,*)'INPUT DURATION OF RUN TIME (hours) '
      write(6,*)'PRESENT VALUE IS ',immax
      k1=1
      read(5,*,iostat=k1,err=305)immax
      if(jr.gt.0)goto 435
c
  320 write(6,*)'WHAT NAME DO YOU WISH TO SAVE THIS FILE UNDER'
      k1=1
      answ='y'
      read(5,*,iostat=k1,err=320)name
      inquire(file=name,exist=fexist)
      if(fexist)call bmsg2(answ)
      if(answ.ne.'y')goto 320
      open(10,file=name)
      rewind 10
      k=10
      call bview
      close(10)
  321 write(6,*)'INPUT ANY NUMBER TO CONTINUE'
      read(5,*,iostat=k1,err=321)icheck
      goto 1
c
c
c THIS SECTION ALLOWS YOU TO CHANGE A PARAMETFR VALUE
c
c
```

```
 430  write(6,*)'INPUT NAME OF FILE TO BE READ'
      k1=1
      read(5,*,iostat=k1,err=430)name
      inquire(file=name,exist=fexist)
      if(.not.fexist)call bmsg1
      if(.not.fexist)goto 430
      open(2,file=name)
      rewind 2
      k=2
      call bread1
      call bread2
      call bread3
      close(2)
      iset=1
      k=6
      call bview
 435  write(6,*)'PLEASE SELECT THE IDENTIFYING VALUE OF THE PARAMETER'
      write(6,*)'YOU WISH TO CHANGE FROM THE FOLLOWING SERIES OF LISTS'
      write(6,*)'  '
      write(6,*)'1.METEOROLOGICAL DATA'
      write(6,*)'2.PRECIPITATION DATA'
      write(6,*)'3.SOILS DATA'
      write(6,*)'4.RUN DEFINITION DATA'
      write(6,*)'5.SAVE FILE'
      write(6,*)'6.EXIT TO MAIN MENU'
      k1=1
      read(5,*,iostat=k1,err=435)kr
      goto(501,502,503,504,320,1)kr
      if(kr.gt.4)goto 1000
c
 501  write(6,*)'CHOOSE FROM THE FOLLOWING PARAMETERS'
c
      write(6,*)'1.PRECIPITATION VALUES'
      write(6,*)'2.RAIN START TIME'
      write(6,*)'3.RAIN FINISH TIME'
      write(6,*)'4.CLOUD TYPE'
      write(6,*)'5.BAROMETRIC PRESSURE'
      write(6,*)'6.ALBEDO OF SURFACE'
      write(6,*)'7.NUMBER OF AIR TEMPERATURE OBSERVATIONS'
      write(6,*)'8.AIR TEMPERATURE OBSERVATIONS'
      write(6,*)'9.NUMBER OF RELATIVE HUMIDITY OBSERVATIONS'
      write(6,*)'10.RELATIVE HUMIDITY OBSERVATIONS'
      write(6,*)'11.NUMBER OF CLOUD COVER OBSERVATIONS'
      write(6,*)'12.CLOUD COVER OBSERVATIONS'
      write(6,*)'13.NUMBER OF GROUND TEMP OBSERVATIONS'
      write(6,*)'14.GROUND TEMP OBSERVATIONS'
      write(6,*)'15.NUMBER OF WIND SPEED OBSERVATIONS'
      write(6,*)'16.WIND SPEED OBSERVATIONS'
      write(6,*)'17.***EXIT***'
c
c
      k1=1
      read(5,*,iostat=k1,err=501)jr
      goto(502,295,300,50,55,71,120,198,125,215,130,
     & 231,140,241,145,251,435)jr
       goto 502
c
 502  write(6,*)'CHOOSE FROM THE FOLLOWING PARAMETERS'
```

```
      write(6,*)'1. NUMBER OF HOURS OF RAIN'
      write(6,*)'2. ALL PRECIPITATION VALUES'
      write(6,*)'3. ONE PRECIPITATION VALUE'
      write(6,*)'4. RAIN START TIME'
      write(6,*)'5. RAIN STOP TIME'
      write(6,*)'6. EXIT TO MENU'
c
      k1=1
      read(5,*,iostat=k1,err=502)jr
      goto(285,286,505,295,300,435)jr
      goto 502
c
  505 write(6,*)'WHICH HOUR OF PRECIPITATION DO YOU WISH TO ALTER?'
      k1=1
      read(5,*,iostat=k1,err=505)i
      write(6,*)' INPUT PRECIPITATION (m) FOR HOUR ',i
      write(6,*)'PRESENT VALUE IS ',ppt(i)
      k1=1
      read(5,*,iostat=k1,err=505)ppt(i)
      goto 502
c
  503 write(6,*)'CHOOSE FROM THE FOLLOWING PARAMETERS'
      write(6,*)'1. INTERNAL ANGLES OF FRICTION, PHI AND PHIB'
      write(6,*)'2. SOIL COHESION'
      write(6,*)'3. SOIL SHEAR MODULUS'
      write(6,*)'4. SOIL DENSITY'
      write(6,*)'5. DEPTH OF CONE TIP'
      write(6,*)'6. NUMBER OF LAYERS'
      write(6,*)'7. SOIL TYPES OF LAYERS'
      write(6,*)'8. ORGANIC MATTER OF LAYERS'
      write(6,*)'9. MOISTURE CONTENT OF LAYERS'
      write(6,*)'10. THICKNESS OF LAYERS'
      write(6,*)'11. SLOPE ANGLE'
      write(6,*)'12. ZENITH ANGLE'
      write(6,*)'13. SURFACE AZIMUTH ANGLE'
      write(6,*)'14. LATITUDE OF SITE'
      write(6,*)'15. TOPOGRAPHIC A/S VALUE'
      write(6,*)'16. EXIT TO MENU'
c
      k1=1
      read(5,*,iostat=k1,err=503)jr
      goto(400,405,410,415,420,10,15,27,276,289,40,45,60,65,11,435)jr
      goto 503
c
 504  write(6,*)'CHOOSE FROM THE FOLLOWING PARAMETERS'
c
      WRITE(6,*)'1. ITERATION TIME INTERVAL'
      write(6,*)'2. DURATION OF RUN TIME'
      write(6,*)'3. JULIAN DAY'
      write(6,*)'4. EXIT TO MENU'
c
      k1=1
      read(5,*,iostat=k1,err=504)jr
      goto(275,305,70,435)jr
      goto 504
c
c
  100 format(1x,'THIS PART OF THE PACKAGE SETS UP AN INPUT FILE')
  110 FORMAT(/1X,'DO YOU WISH TO CONTINUE? y OR n')
c
c
 1010 stop
 1000 return
      end
```

```
c
c********************************************************
c
c SUBROUTINE     BMSG
c
c********************************************************
c
      subroutine bmsg(answ)
c
c
c This subroutine sends a message to the user when
c a file (1) does not exist and the system has been
c told by the user to read it, and (2) that a file
c already exists when wanting to over write.  The
c system asks for confirmation of overwriting.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c January, 1988.
c
c
c 'bmsg' is called by 'bcntl'
c                     'bmenu'
c                     'rrcid'
c                     'rread'
c                     'rsave'
c                     'tcntl'
c                     'tred'
c                     'vcntl'
c                     'vmenu'
c
c
      character answ*1
      entry bmsg1
c
c
      write(6,*)'    '
      write(6,*)'FILE DOES NOT EXIST'
      write(6,*)'    '
c
      return
c
c
      entry bmsg2(answ)
c
      write(6,*)'   '
      write(6,*)'FILE ALREADY EXISTS - DO YOU WISH TO OVERWRITE?'
      write(6,*)'  '
      write(6,*)' y OR n '
      write(6,*)'   '
      read(5,*)answ
      return
      end
```

```
c
c********************************************************
c
c SUBROUTINE    BREAD
c
c********************************************************
c
      subroutine bread
c
c This subroutine reads the input data for the BRISTOL SOIL
c STRENGTH SCHEME.
c Created by J.E. Cochrane
c at Geography Department, University of Bristol, England.
c September, 1987.
c
c
c 'bread' has 3 entry points - 'bread1', 'bread2', & 'bread3'
c
c 'bread1' is called by 'bmenu' when editing, otherwise 'bcntl'
c 'bread2' is called by 'bmenu' when editing, otherwise 'bcntl'
c 'bread3' is called by 'bmenu' when editing, otherwise 'binfil'
c
c
      include 'spec.bone'
      include 'spec.btwo'
      include 'spec.bthree'
      include 'spec.bgrnd'
c
c
c
      entry bread1
       write(14,*)' read1'
      k=2
      read(k,*)nos,aos
      read(k,*)isty(1),isty(2),isty(3)
      read(k,*)op(1),op(2),op(3)
      return
c
c
      entry bread2
      write(14,*)' read2'
      k=2
c
      read(k,*)max(1),max(2),max(3),max(4),max(6)
      read(k,*)slope,za,ncloud,press,surfacaz,lat,days
      do 10 j=1,6
         if(j.eq.5)goto 20
         read(k,*)(yyy(i,j),i=1,max(j))
         read(k,*)(xxx(i,j),i=1,max(j))
   20    CONTINUE
   10 CONTINUE
c
      read(k,*)nl
      read(k,*)detcap,af
      do 30 i=1,1
         read(k,*)(tf(i,j),j=1,nl)
         write(14,*)(tf(i,j),j=1,nl)
   30 CONTINUE
```

```
c
      read(k,*)(tcom(i),i=1,nl)
      read(k,*)ir
      read(k,*)(ppt(i),i=1,ir)
c
      return
c
c
      entry bread3
      write(14,*)' read3'
      k=2
      kl=1
c
c
      read(k,*,iostat=kl,err=100)alr,amr
      read(k,*,iostat=kl,err=100)(nla(i),i=1,nos),immax
      read(k,*,iostat=kl,err=100)alb
      read(k,*,iostat=kl,err=100)phi,c,phib,gg,zz
      k=6
  100 CONTINUE
c
      aal=1.48
      d=0.799
      alpha=15.0
      gamma=0.067
      ua=-17.0
c
      return
c
      end
```

```
c
c**********************************************************
c
c SUBROUTINE     BSCHARS
c
c**********************************************************
c
      subroutine bschars
c
c
c This subroutine accesses the soil information data bank in the
c BRISTOL SOIL STRENGTH SCHEME.
c Created by J.E. Cochrane
c at Geography Department, University of Bristol, England.
c July, 1987.
c
c
c Key:- soil type
c
c
c 'bschars' is called by 'bcntl'
c
      include 'spec.bnow'
      include 'spec.bgrnd'
c
      include 'spec.bdl'
      include 'spec.bdata'
c
c
      open(15,file='check.d')
      write(14,*)' scars'
      write(14,*)j
c
c
      bsatcon=log10(asatcon(isty(j)))*aos
      satcon(j)=g05ddf(bsatcon,ssatcon(isty(j)))
      satcon(j)=10**satcon(j)
c
      bp(j)=g05ddf(abp(isty(j)),sbp(isty(j)))
      sr(j)=g05ddf(asr(isty(j)),ssr(isty(j)))
      bdm(j)=g05ddf(am(isty(j)),sm(isty(j)))
      sp(j)=g05ddf(asp(isty(j)),ssp(isty(j)))
      cp(j)=g05ddf(acp(isty(j)),scp(isty(j)))
c
      write(15,*)' satcon',j,satcon(j)
      write(15,*)' bp',j,bp(j)
      write(15,*)' sr',j,sr(j)
      write(15,*)' bdm',j,bdm(j)
      write(15,*)' sp',j,sp(j)
      write(15,*)' cp',j,cp(j)
c
c
      return
      end
```

```
c
c*********************************************************
c
c SUBROUTINE    BSMCURV
c
c*********************************************************
c
      subroutine bsmcurv
c
c
c This subroutine calculates the new 20 point suction moisture curve
c in the BRISTOL SOIL STRENGTH SCHEME.
c Original by S. Howes
c Adapted by J.E. Cochrane
c at the Geography Department, University of Bristol.
c
c
c 'bsmcvurv' is called by 'bcntl'
c
      include 'spec.bgrnd'
      include 'spec.bman'
C
C
C Calculate gradients of this new suction-moisture curve
c
      nq=10
      nnq=NQ-1
      do 200 I=1,nnq
 200  ag(j,i)=(y(j,i+1)-y(j,i))/(ax(j,i+1)-ax(j,i))
C
C Calculate max and min moisture values, and determine the size of
C equal intervals.
C
      do 305 i=1,10
 305  x(j,i)=ax(j,i)
      xmax=ax(j,10)
      xmin=ax(j,1)
      xint=(xmax-xmin)/19.
C
C Determine the new values of moisture-equal intervals
C
      xnew(j,1)=xmin
      do 300 i=2,19
300   xnew(j,i)=xnew(j,1)+(xint*(i-1))
      xnew(j,20)=xmax
C
C Determine the associated new values of suction
C
      do 350 i=1,19
        do 400 k=1,nnq
         if(xnew(j,i).ge.ax(j,k).and.xnew(j,i).lt.ax(j,k+1))
     & ynew(j,i)=y(j,k)+ag(j,k)*(xnew(j,i)-ax(j,k))
400     CONTINUE
350   CONTINUE
      ynew(j,20)=y(j,nq)
C
C
C
```

```
c
      return
      end
```

```
c
c*********************************************************
c
c SUBROUTINE    BSOILBD
c
c*********************************************************
c
       subroutine bsoilbd
c
c
c This subroutine calculates soil bulk density (sbd) and the basic
c suction moisture curve for given suctions in the BRISTOL SOIL
c STRENGTH SCHEME.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c February, 1987.
c
c
c 'bsoilbd' is called by 'bcntl'
c
c
      include 'spec.bgrnd'
      include 'spec.bman'
      include 'spec.bone'
      include 'spec.bextra'
      include 'spec.bdat1'
c
c
      sbd(j)=(100/((op(j)/0.224)+((100-op(j))/bdm(j))))
      do 10 i=1,9
      aax(j,i)=(aa(i)+(bb(i)*sp(j))+(cc(i)*cp(j))+(dd(i)*op(j)
     & )+(ee(i)*sbd(j)))
      write(14,*)'aax(j,i)',aax(j,i)
   10 CONTINUE
c
      return
      end
```

```
c
c*************************************************************
c
c SUBROUTINE    BVIEW
c
c*************************************************************
c
      subroutine bview
c
c
c This subroutine will printout an input file for the BRISTOL SOIL
c STRENGTH SCHEME to either a datafile or the screen.
c Created by J.E. Cochrane
c at Geography Department, University of Bristol, England.
c October, 1987.
c
c
c 'bview' is called by 'bmenu'
c
      include 'spec.bone'
      include 'spec.btwo'
      include 'spec.bthree'
      include 'spec.bgrnd'
c
c
      write(k,*)nos,aos
      write(k,*)(isty(i),i=1,nos)
      write(k,*)(op(i),i=1,nos)
      write(k,*)(max(i),i=1,4),max(6)
      write(k,*)slope,za,ncloud,press,surfcaz,lat,days
      write(k,*)(yyy(i,1),i=1,max(1))
      write(k,*)(xxx(i,1),i=1,max(1))
      write(k,*)(yyy(i,2),i=1,max(2))
      write(k,*)(xxx(i,2),i=1,max(2))
      write(k,*)(yyy(i,3),i=1,max(3))
      write(k,*)(xxx(i,3),i=1,max(3))
      write(k,*)(yyy(i,4),i=1,max(4))
      write(k,*)(xxx(i,4),i=1,max(4))
      write(k,*)(yyy(i,6),i=1,max(6))
      write(k,*)(xxx(i,6),i=1,max(6))
c
      write(k,*)nl
      write(k,*)detcap,af
      write(k,*)(tf(1,i),i=1,nl)
      write(k,*)(tcom(i),i=1,nl)
      write(k,*)ir
      write(k,*)(ppt(i),i=1,ir)
      write(k,*)alr,amr
      write(k,*)(nla(i),i=1,nos),immax
      write(k,*)alb
      write(k,*)phi,c,phib,gg,zz
c
      return
      end
```

```
c
      write(6,*)'****************************************************'
      write(6,*)'WELCOME TO THE BRISTOL ROUTE PACKAGE'
      write(6,*)'****************************************************'
   10 write(6,*)'ENTER ANY NUMBER TO CONTINUE'
      k1 =1
      read(5,*,iostat=k1,err=10)fax
   20 write(6,*)'MASTER MENU'
      write(6,*)' 1. ACCESS BRISTOL RCI PROGRAM'
      write(6,*)' 2. ACCESS VSAS2 RCI PROGRAM'
      write(6,*)' 3. OPTIMUM ROUTE PROGRAM'
      write(6,*)' 4. EVALUATION OF RCI OVER A GIVEN ROUTE'
      write(6,*)' 5. FINISH'
      write(6,*)'INPUT OPTION 1-5'
c
c
      k1=1
      read(5,*,iostat=k1,err=20)io
      if(io.gt.5.or.io.lt.1)goto 20
      goto(30,40,50,55,60)io
   30 call bcntl
      goto 20
   40 call vcntl
      goto 20
   50 call rcntl
      goto 20
   55 call tcntl
      goto 20
   60 stop
      end
```

```
c
c*********************************************************
c
c SUBROUTINE   RCHECK
c
c*********************************************************
c
      subroutine rcheck
c
c
c This subroutine allows the user to edit a grid file for use in the
c GRID CREATE SCHEME.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c September, 1987.
c
c
c 'rcheck' is called by 'rcntl'
c
      include 'spec.rgrid'
c
c
      write(6,132)
      write(6,135)(i,i=1,ii)
      write(6,136)
      do 20 j=1,jj
         write(6,140)j,(ig(j,i),i=1,ii)
   20 CONTINUE
  600 write(6,150)
      k10=3
      read(5,*,iostat=k10,err=800)ans
      if (ans.ne."y")goto 1000
      write(6,160)
      k11=2
      read(5,*,iostat=k11,err=900)j
      write(6,170)
      k12=2
      k13=7
      read(5,*,iostat=k12,err=900)i
      write(6,180)ig(j,i)
      read(5,*,iostat=k13,err=900)ig(j,i)
      write(6,190)
      write(6,132)
      write(6,135)(i,i=1,ii)
      write(6,136)
      do 620 j=1,jj
      write(6,140)j,(ig(j,i),i=1,ii)
  620 CONTINUE
      goto 300
  800 write(6,*)'  YOU HAVE INPUT A NUMERIC INSTEAD OF y OR n'
      goto 600
  900 write(6,*)' YOU HAVE INPUT A NON-INTEGER VALUE - TRY AGAIN'
      goto 600
c
 1000 if(ans.ne."n")goto 600
  100 format(1x,' HOW MANY COLUMNS? ')
  110 format(1x,' HOW MANY ROWS? ')
  120 format(1x,2i3)
  130 format(1x,' FOR ROW ',i3,' COLUMN ',i3,'
     & ENTER IDENTIFYING NUMBER ')
  132 format(15x,'COLUMN')
  135 format(12x,15i3)
  136 format(/,1x,' ROW',/)
  140 format(1x,i3,8x,10i3)
  150 format(1x,'DO YOU WISH TO CHANGE ANY GRID VALUE? Y or N')
  160 format(1x,'ENTER ROW NUMBER')
  170 format(1x,'ENTER COLUMN NUMBER')
  180 format(1x,'THE VALUE OF THIS CELL IS',i3,/,
     &'PLEASE ENTER NEW IDENTIFYING VALUE')
  190 format(1x,'THE NEW GRID IS AS FOLLOWS')
c
  300 return
      end
```

```
c
c**********************************************************
c
c SUBROUTINE    RCNTL
c
c**********************************************************
c
      subroutine rcntl
c
c This subroutine controls the creation, editing and saving
c of a grid.  It also calls the optimum route model.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c September, 1987.
c
c
c 'rcntl' is called by 'link'
c
c 'rcntl' calls 'rcreate'
c 'rcntl' calls 'rcheck'
c 'rcntl' calls 'rsave'
c 'rcntl' calls 'rread'
c 'rcntl' calls 'rroute'
c
c
c
c
      include 'spec.rgrid'
      include 'spec.rrci'
c
   10 write(6,100)
      write(6,110)
      write(6,120)
      write(6,130)
      write(6,140)
      write(6,150)
      write(6,155)
      write(6,160)
      k=1
      read(5,*,iostat=k1,err=10)ino
c
      if(ino.gt.6)goto 10
      if(ino.lt.1)goto 10
      if (ino.eq.1)call rcreate
      if (ino.eq.2)call rcheck
      if (ino.eq.3)call rsave
      if (ino.eq.4)call rread
      if (ino.eq.5)call rroute
      if (ino.eq.6)goto 20
      goto 10
c
c
  100 format(1x,'*** MENU SELECTION FOR GRID SET-UP ***')
  110 format(/,1x,'1. CREATE GRID')
  120 format(1x,'2. CHECK GRID')
  130 format(1x,'3. SAVE GRID')
  140 format(1x,'4. READ EXISTING GRID')
  150 format(1x,'5. EXIT TO ROUTE SELECTION')
  155 format(1x,'6. QUIT PROGRAM')
  160 format(//,1x,'PLEASE ENTER SELECTION 1-6')
c
c
   20 end
```

```
c
c*********************************************************
c
c SUBROUTINE    RCREATE
c
c*********************************************************
c
      subroutine rcreate
c
c
c This subroutine creates a grid to the users specifications.
c This is used in the GRID CREATE SCHEME.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c September, 1987.
c
c
c 'rcreate' is called by 'rcntl'
c
      include 'spec.rgrid'
c
c
      if(kit.ge.99)goto 700
   10 write(6,100)
      k1=2
      read(5,*,iostat=k1,err=800)ii
      write(6,110)
      k2=2
      read(5,*,iostat=k2,err=800)jj
c
      do 500 j=1,jj
         do 400 i=1,ii
         write(6,130)j,i
         k1=3
         read(5,*,iostat=k1,err=800)ig(j,i)
  400    CONTINUE
  500 CONTINUE
c
      write(6,140)
      write(6,150)(i,i=1,ii)
      write(6,160)
      do 520 j=1,jj
        write(6,170)j,(ig(j,i),i=1,ii)
  520 CONTINUE
      kit=999
c
c
  100 format(1x,' HOW MANY COLUMNS?')
  110 format(1x,'HOW MANY ROWS?')
  120 format(1x,2i3)
  130 format(1x,' FOR ROW ',i3,' COLUMN ',i3,' ENTER ID')
  140 format(15x,'COLUMN')
  150 format(12x,15i3)
  160 format(/,1x,' ROW',/)
  170 format(1x,i3,8x,15i3)
  710 format(1x,'THERE IS ALREADY A GRID CREATED')
  720 format(1x,'DO YOU WISH TO OVERWRITE? y/n')
c
```

```
c
      if(kit.ge.99)goto 20
  700 write(6,710)
      write(6,720)
      read(5,*)ans
      if(ans.eq.'y')goto 10
      return
  800 write(6,*)'  YOU HAVE INPUT A NON-INTEGER VALUE'
   20 return
      end
```

```
c
c********************************************************
c
c SUBROUTINE     RPRINT
c
c********************************************************
c
      subroutine rprint
c
c
c
c This subroutine prints out the evaluation of RCI over
c a given route.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c September, 1987.
c
c 'rprint' is called by 'rrcid'
c
      include 'spec.rgrid'
      include 'spec.rrci'
      include 'spec.rtrav'
c
c
      entry rprint1
c
c
      open(10,file='route.out')
      do 10 j=1,10
      write(6,100)j,trackc(j),trackr(j)
   10 CONTINUE
  100 format(1x,3i5)
      return
      end
```

```
c
c*********************************************************
c
c SUBROUTINE    RRCID
c
c*********************************************************
c
      subroutine rrcid
c
c
c This subroutine calculates an optimum route from a specified
c grid and library of data files.  It asks for start and finish
c grid squares.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c September, 1987.
c
c
c 'rrcid' is called by 'rroute'
c
c 'rrcid' calls 'bmsg1'
c 'rrcid' calls 'rprint1'
c
c
      include 'spec.rgrid'
      include 'spec.rrci'
      include 'spec.rtrav'
      dimension check(15,15)
      dimension rtc(30)
      dimension rtr(30)
c
      integer posc,posr,br,bc,biasc,biasr,movec,mover
      integer check,lim1
      integer diff,total
      integer cs,cf,rs,rf
      character any*8
      character fname*20
      logical fexist
c
    5 write(6,*)'INPUT NAME OF GRID FILE'
      read(5,*)fname
      inquire(file=fname,exist=fexist)
      if(.not.fexist)call bmsg1
      if(.not.fexist)goto 5
      open(11,file=fname)
c
c lowest limit of rci value allowed is set by lim1
c
      lim1=45
c
      write(6,*)'jj',jj
      write(6,*)'ii',ii
      do 10 j =1,jj
         do 20 i=1,ii
            igg(i,j)=ir(ig(j,i))
            check(j,i)=0
   20    CONTINUE
   10 CONTINUE
```

```
c
      do 30 j=1,jj
        write(11,100)j,(igg(i,j),i=1,ii)
  30  CONTINUE
c
c
      write(6,200)
      write(6,210)
      k1=1
      read(5,*,iostat=k1,err=400)cs
      write(6,*)cs
      write(6,220)
      k1=1
      read(5,*,iostat=k1,err=400)rs
      write(6,*)rs
      write(6,230)
      write(6,210)
      k1=1
      read(5,*,iostat=k1,err=400)cf
      write(6,*)cf
      write(6,220)
      k1=1
      read(5,*,iostat=k1,err=400)rf
      write(6,*)rf
c
c start
c
      posc=cs
      posr=rs
c
      do 150 kkk=1,30
      rtc(kkk)=posc
      rtr(kkk)=posr
      trackc(kkk)=posc
      trackr(kkk)=posr
c
      biasc=cf-posc
      biasr=rf-posr
      write(11,*)biasc,biasr,'bias'
c
      if(posc.eq.cf.and.posr.eq.rf)call rprint1
      if(posc.eq.cf.and.posr.eq.rf)return
c
      if(biasc.lt.0)bc=-1
      if(biasc.gt.0)bc=1
      if(biasc.eq.0)bc=0
c
      write(6,*)biasc
      if(biasr.lt.0)br=-1
      if(biasr.gt.0)br=1
      if(biasr.eq.0)br=0
c
      write(6,*)biasr
c
      iac=(iabs(biasc))
      iar=(iabs(biasr))
      total=iar+iac
      diff = (max0(iac,iar)+1)
      write(6,*)iac,iar,diff,total,'iac, iar, dif, tot'
c
      if(biasr.eq.0)goto 400
      if(biasc.eq.0)goto 280
c
c
      mover=posr
      movec=posc
      if(check(posc+bc,posr).gt.0)movec=posc+bc
      if(check(posc+bc,posr).gt.0.and.igg(posc+bc,posr).gt.liml)goto 85
c
      if(check(posc,posr+br).gt.0)mover=posr+br
      if(check(posc,posr+br).gt.0.and.igg(posc,posr+br).gt.liml)goto 90
c
      if(check(posc+bc,posr+br).gt.0)movec=posc+bc
      if(check(posc+bc,posr+br).gt.0)mover=posr+br
      if(check(posc+bc,posr+br).gt.0.and.igg(posc+bc,posr+br).gt.liml)
     & goto 95
c
      check(posc+bc,posr)=kkk
      check(posc,posr+br)=kkk
      check(posc+bc,posr+br)=kkk
c
      kan=igg(posc+bc,posr)*iac/total
      kbn=igg(posc,posr+br)*iar/total
      kcn=igg(posc+bc,posr+br)*diff/total
c
      if(kan.gt.kbn.and.kan.gt.kcn)movec=posc+bc
      if(kan.gt.kbn.and.kan.gt.kcn)mover=posr
c
      if(kbn.gt.kan.and.kbn.gt.kcn)movec=posc
      if(kbn.gt.kan.and.kbn.gt.kcn)mover=posr+br
c
      if(kcn.ge.kan.and.kcn.ge.kbn)movec=posc+bc
      if(kcn.ge.kan.and.kcn.ge.kbn)mover=posr+br
c
      write(6,*)'kan',kan,igg(posc+bc,posr)
      write(6,*)'kbn',kbn,igg(posc,posr+br)
      write(6,*)'kcn',kcn,igg(posc+bc,posr+br)
      write(6,*)'posc',posc,movec
      write(6,*)'posr',posr,mover
      goto 310
c
   85 kkk=check(posc+bc,posr)
      goto 310
   90 kkk=check(posc,posr+br)
      goto 310
   95 kkk=check(posc+bc,posr+br)
      goto 310
c
c
c N-S movt.
c
  280 CONTINUE
c
c
      write(6,*)check(posc,posr+br),check(posc+1,posr+br)
     & ,check(posc-1,posr+br)
```

```
      mover=posr+br
      movec=posc
      if(check(posc,posr+br).gt.0.and.igg(posc,posr+br).gt.liml)goto 300
      if(check(posc+1,posr+br).gt.0)movec=posc+1
      if(check(posc+1,posr+br).gt.0.and.igg(posc+1,posr+br).gt.liml)
     & goto 301
      if(check(posc-1,posr+br).gt.0)movec=posc-1
      if(check(posc-1,posr+br).gt.0.and.igg(posc-1,posr+br).gt.liml)
     & goto 302
      check(posc,posr+br)=kkk
      check(posc+1,posr+br)=kkk
      check(posc-1,posr+br)=kkk
      write(6,*)check(posc,posr+br),check(posc+1,posr+br)
     & ,check(posc-1,posr+br)
      if(posc.eq.cf.and.posr+br.eq.rf)goto 80
      if(posc+1.eq.cf.and.posr+br.eq.rf)goto 80
      if(posc-1.eq.cf.and.posr+br.eq.rf)goto 80
c
      ian = igg(posc,posr+br)
      ibn = (igg(posc+1,posr+br)*(iar-1)/total)
      icn = igg(posc-1,posr+br)*(iar-1)/total
      write(6,*)posc,posr,'col, row'
      write(6,*)igg(posc,posr+br),'ian',br
      write(6,*)igg(posc+1,posr+br),'ibn'
      write(6,*)igg(posc-1,posr+br),'icn'
      write(6,*)ian,ibn,icn
      write(6,*)posc,posr
c
c
c comparison of next potential grid squares (3)
c
c
      if(ian.gt.ibn)movec=posc
      if(ibn.ge.ian)movec=posc+1
      if(ian.lt.ibn)ian=ibn
      if(icn.gt.ian)movec=posc-1
      mover=posr+br
      ian=0
      ibn=0
      icn=0
c
c
c set new position according to move decision
c
c
      goto 310
  300 kkk = check(posc,posr+br)
      goto 310
  301 kkk = check(posc+1,posr+br)
      goto 310
  302 kkk = check(posc-1,posr+br)
  310 CONTINUE
      posc=movec
      posr=mover
      write(6,*)kkk,posc,posr
      read(5,*)any
      goto 150
  400 CONTINUE
```

```
c     W-E movt.
c
      if(posr.eq.rf.and.posc+bc.eq.cf)goto 80
      if(posr+br.eq.rf.and.posc+bc.eq.cf)goto 80
      if(posr-br.eq.rf.and.posc+bc.eq.cf)goto 80
c
      movec=posc+bc
      if(check(posc+bc,posr).gt.0)mover=posr
      if(check(posc+bc,posr).gt.0.and.igg(posc+bc,posr).gt.lim1)goto 115
c
      if(check(posc+bc,posr+1).gt.0)mover=posr+1
      if(check(posc+bc,posr+1).gt.0.and.igg(posc+bc,posr+1).gt.lim1)
     & goto 125
c
      if(check(posc+bc,posr-1).gt.0)mover=posr-1
      if(check(posc+bc,posr-1).gt.0.and.igg(posc+bc,posr-1).gt.lim1)
     & goto 135
c
      check(posc+bc,posr)=kkk
      check(posc+bc,posr+1)=kkk
      check(posc+bc,posr-1)=kkk
c
      jan=igg(posc+bc,posr)
      jbn=igg(posc+bc,posr+1)*iac/total
      jcn=igg(posc+bc,posr-1)*iac/total
      write(6,*)'jan',jan,jbn,jcn
      write(6,*)posc,posr
c
c
c
c     comparison of next 3 grid squares in W-E direction
c
c
c
c
      if(jan.ge.jbn)mover=posr
      if(jbn.ge.jan)mover=posr+1
      if(jan.lt.jbn)jan=jbn
      if(jcn.gt.jan)mover=posr-1
      movec=posc+bc
c
c
      goto 310
  115 kkk=check(posc+bc,posr)
      goto 310
  125 kkk=check(posc+bc,posr+1)
      goto 310
  135 kkk=check(posc+bc,posr-1)
      goto 310
  150 CONTINUE
   80 trackc(kkk+1)=cf
      trackr(kkk+1)=rf
      call rprint1
  100 format(1x,i3,8x,15i3)
  200 format(1x,'DEFINE STARTING POSITION')
  210 format(/,1x,'ENTER COLUMN')
  220 format(/,1x,'ENTER ROW')
  230 format(//,1x,'DEFINE FINISH POSITION')
```

```
c
c
      return
c
c
      end
```

```
c
c**********************************************************
c
c SUBROUTINE    RREAD
c
c**********************************************************
c
      subroutine rread
c
c
c This subroutine reads a grid file in the GRID CREATE SCHEME.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c September, 1987.
c
c
c 'rread' is called by 'rcntl'
c
      include 'spec.rgrid'
      include 'spec.rrci'
c
c
c
c
      character col*8,row*3
      logical fexist
      integer t
      include 'spec.tset'
c
      entry rread1
c
c
      if(kit.ge.99)goto 700
   10 write(6,100)
      read(5,*)name
      inquire(file=name,exist=fexist)
      if(.not.fexist)call bmsg1
      if(.not.fexist)goto 10
      open(3,file=name)
      kit=999
      k1=2
      read(3,*,iostat=k1,err=20,end=1000)ii,jj
      read(3,*)col
      read(3,*)(ia(i),i=1,ii)
      read(3,*)row
      write(6,150)
      write(6,170)
      write(6,160)(i,i=1,ii)
      write(6,165)
      do 200 j=1,jj
      read(3,*)ij,(ig(j,i),i=1,ii)
      write(6,180)ij,(ig(j,i),i=1,ii)
  200 CONTINUE
      if(kit.ge.99)goto 20
  700 write(6,150)
      write(6,170)
      write(6,160)(i,i=1,ii)
      do 210 j=1,jj
```

```
         write(6,180)j,(ig(j,i),i=1,ii)
  210 CONTINUE
      write(6,190)
      read(5,*)ans
      if(ans.eq.'y')goto 10
      goto 20
 1000 write(6,300)
   20 return
c
c
  100 format(1x,'NAME OF FILE TO BE RETRIEVED?')
  130 format(12x,15i3,/)
  140 format(1x,i3,8x,15i3)
  150 format(15x,'COLUMN')
  160 format(12x,15i3,//)
  165 format(1x,25x)
  170 format(/,1x,' ROW',/)
  180 format(1x,i3,8x,15i3)
  190 format(1x,'DO YOU WISH TO OVERWRITE EXISTING GRID? y/n')
  300 format(1x,'FILE DOES NOT EXIST')
c
      entry rread2
c
c
      write(6,9100)
      write(6,9110)
      write(6,9120)
      read(5,*)ans1
      if(ans1.eq.'y')return
      write(6,9130)
      read(5,*)t
      write(6,9140)
      read(5,*)n
c
      do 910 k=1,n
         name=try(k)
         open(4,file=name)
         rewind 4
         write(6,241)name
         do 920 kk=1,t-1
            read(4,*)iab
  920    CONTINUE
         read(4,*)ir(k)
  910 CONTINUE
c
c
  241 format(1x,15A)
  888 format(1x,1A)
 9100 format(1x,'IT IS ASSUMED THAT THERE ARE DATA FILES EXISTING')
 9110 format(1x,'NAMED data.n, WHERE n IS THE ID IN THE GRID')
 9120 format(1x,'DO YOU WISH TO RETURN TO MAIN MENU? y/n')
 9130 format(1x,'PLEASE INPUT TIME REQUIRED (AS HOURS FROM START)')
 9140 format(1x,'PLEASE DEFINE MAX. NUMBER OF FILES TO BE ACCESSED')
c
c
      return
      end
```

```
c
c*********************************************************
c
c SUBROUTINE    RROUTE
c
c*********************************************************
c
      subroutine rroute
c
c
c This subroutine allows the user to escape from the
c route management scheme if a full library of data
c files does not exist.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c September, 1987.
c
c 'rroute' is called by 'rcntl'
c
c 'rroute' calls 'read2'
c 'rroute' calls 'rrcid'
c
c
      include 'spec.rgrid'
      include 'spec.rrci'
c
c
      call rread2
      if(ans1.eq.'y')return
      call rrcid
      return
      end
```

```
c
c********************************************************
c
c SUBROUTINE    RSAVE
c
c********************************************************
c
      subroutine rsave
c
c
c This subroutine saves the grid with identifiers in the GRID CREATE
c SCHEME.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c September, 1987.
c
c
c ’rsave’ is called by ’rcntl’
c
      logical fexist
      character answ*1
c
c
      include ’spec.rgrid’
c
c
c
c
      write(6,200)
      read(5,*)ans
      if(ans.ne.’y’)write(6,210)
      if(ans.ne.’y’)read(5,*)ans
      if(ans.ne.’y’)goto 670
      write(6,220)
   10 read(5,*)name
      answ=’y’
      inquire(file=name,exist=fexist)
      if(fexist)call bmsg2(answ)
      if(answ.ne.’y’)goto 10
      open(3,file=name)
      write(3,135)ii,jj
      write(3,140)
      write(3,150)(i,i=1,ii)
      write(3,160)
      do 300 j=1,jj
         write(3,170)j,(ig(j,i),i=1,ii)
  300 CONTINUE
      write(3,180)
      close(3)
c
  670 CONTINUE
c
c
c
  135 format(1x,2i5)
  140 format(15x,’COLUMN’)
  150 format(12x,15i3)
  160 format(1x,’ ROW’)
  170 format(1x,i3,8x,15i3)
  180 format(1x,'999')
  200 format(1x,'DO YOU WISH TO SAVE THIS GRID?')
  210 format(1x,'A SECOND CHANCE TO SAVE THE GRID Y or N?')
  220 format(1x,'PLEASE TYPE IN NAME OF FILE THAT GRID IS TO',/,
     &'SAVED IN')
c
      return
      end
```

```
c
c**********************************************************
c
      subroutine tcntl
c
c
c This subroutine controls the evaluation of RCI over
c a specified route.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c October, 1987.
c
c
c 'tcntl' is called by 'link'
c
c 'tcntl' calls 'tred1'
c               'tred2'
c               'bmsg2'
c
c
      include 'spec.tinput'
c
c
      character name*15
      character answ*1
      logical fexist
      include 'spec.tset'
c
      write(6,400)
      write(6,410)
      write(6,420)
      read(5,*)ik
      if(ik.eq.1)call tred1
      if(ik.eq.2)call tred2
c
      do 20 i=1,n
      write(6,*)i,pos(i),id(i),t(i)
   20 CONTINUE
c
   25 write(6,*)'PLEASE ENTER NAME OF OUTPUT FILE'
      read(5,*)name
      inquire(file=name,exist=fexist)
      if(fexist)call bmsg2(answ)
      if(answ.ne.'y')goto 25
      open(8,file=name)
c
      do 30 i=1,n
      open(3,file=try(id(i)))
         do 40 it=1,(t(i)-1)
            read(3,*)a
   40    CONTINUE
      read(3,*)rci(i)
      write(8,*)i,pos(i),rci(i)
      close(3)
   30 CONTINUE
c
c
c
```

```
  400 format(1x,'PLEASE MAKE CHOICE OF :-')
  410 format(//,1x,'1.CREATE ROUTE')
  420 format(1x,'2.READ ROUTE')
c
c
 1000 CONTINUE
      return
      end
```

```
c
c*******************************************************
c
c SUBROUTINE    TRED
c
c*******************************************************
c
      subroutine tred
c
c
c This subroutine asks the user to specify a route for
c the route evaluation scheme.
c Created by J.E. Cochrane
c at the  Department of Geography, University of Bristol, England.
c October, 1987.
c
c
c ’tred’ is called by ’tcntl’
c
c
      include ’spec.tinput’
c
       character name*10
       character answ*1
       logical fexist
c
      entry tred1
c
      write(6,100)
      read(5,*)n
      do 10 i=1,n
      write(6,110)i
      read(5,*)pos(i)
      write(6,120)i
      read(5,*)id(i)
      write(6,130)i
      read(5,*)t(i)
   10 CONTINUE
c
   15 write(6,140)
      read(5,*)name
      answ=’y’
      inquire(file=name,exist=fexist)
      if(fexist)call bmsg2(answ)
      if(answ.ne.’y’)goto 15
      open(20,file=name)
      rewind 20
      do 20 i =1,n
         write(20,*)i,pos(i),id(i),t(i)
   20 CONTINUE
c
c
  100 format(1x,’INPUT NUMBER OF NODES ON ROUTE’,i3)
  110 format(1x,’INPUT DIST OF NODE ALONG ROUTE’,i3)
  120 format(1x,’INPUT IDENTIFIER OF NODE’i3,)
  130 format(1x,’INPUT TIME STEP OF NODE’i3,)
  140 format(1x,’INPUT NAME OF FILE ROUTE WILL BE SAVED ON’)
c
```

```
      return
c
c
c
      entry tred2
c
c
      write(6,145)
      read(5,*)n
   25 write(6,150)
      read(5,*)name
      inquire(file=name,exist=fexist)
      if(.not.fexist)call bmsg1
      if(.not.fexist)goto 25
      open(20,file=name)
      do 30 i=1,n
         read(20,*)i,pos(i),id(i),t(i)
   30 CONTINUE
c
c
  145 format(1x,'INPUT NUMBER OF NODES IN ROUTE')
  150 format(1x,'INPUT NAME OF FILE TO BE READ')
c
c
       return
       end
```

```
c
c**********************************************************
c
c SUBROUTINE     VBLKVOL
c
c**********************************************************
c
      subroutine vblkvol
c
c Original Bernier, 1982;Whitelaw, 1988.
c
c
c 'vblkvol' is called by 'vcntl'
c
c
      include 'spec.vgeom'
      include 'spec.vphys'
      include 'spec.vindex'
      include 'spec.vdata'
      include 'spec.vcom2'
      include 'spec.vkgr'
c
c     dimension y(5,2,16),elev(5,0:20,5),dep(5,16,5),
c    2          zdist(5),clev(5),alpha(5),diff(5)
      dimension elev(5,0:20,5),alpha(5),diff(5)
c
c
C************SECTION 1
c
c     This section now selects the longest grid network and uses the
c     the lengths derived from its incrementation to divide the other
c     grids. The secondary grids will obviously not fit this
c     perfectly so the nearest fit (longer or shorter) is used.
c
      ni=nl
      nn=ni+1
c
      do 5 k = 1,kno
      ni = nlimk(k)
      cuml = 0.
      x(k,0) = 0.
      x(k,1) = 2 * xi(k,1)
c
      do 10 n=2,ni
            cuml = xi(k,n) - x(k,n-1)
            cuml = 2 * cuml
        zdist = horiz(k,1)
   10   x(k,n) = x(k,n-1) + cuml
   05  CONTINUE
c
c
c-----SECTION 1a-----
c
c
c     This section calculates the angular shifts for the non central
c     flow lines (eg k=1 & k=3 for kno ~ 3 ). hlen is input as the
c     horizontal distance from the stream to the watershed running
c     at 90 deg from the stream.
```

```
c
c
      do 30 k = 1,kno
   30       alpha(k) = asin(hlen(k)/x(k,ni))
c
            ksum = 0
      do 32 k = 1,kno
            ksum = ksum + k
   32 if(k.eq.kno) ik = ksum / kno
c
      do 35 k = 1,kno
            diff(k) = abs(alpha(k)-alpha(ik))
   35 shift(k) = cos(diff(k))
c
c
c-----SECTION 2------
c
c
      do 40 k=1,kno
            ni = nlimk(k)
      do 40 n=1,ni
            avdep(k,n,1) = adepth
            avele(k,n,1) = yi(k,n) - avdep(k,n,1) / 2.
            rdep = depmax(k,n) - avdep(k,n,1)
            sumdep = avdep(k,n,1)
      do 40 j=2,jno
            jx = j-1
            avdep(k,n,j) = bdepth(jx) * rdep
            sumdep = sumdep + avdep(k,n,j)
   40       avele(k,n,j) = yi(k,n) - (sumdep - avdep(k,n,j)/2.)
c
       do 45 k = 1,kno
             ni = nlimk(k)
       do 45 j = 1,jno
       do 44 n = 1,ni-1
             elev(k,n,j) = avele(k,n,j) + ((avele(k,n+1,j) -
     &                     avele(k,n,j))/2.)
   44 CONTINUE
             elev(k,0,j) = avele(k,1,j) - (elev(k,1,j) -
     &                                  avele(k,1,j))
             elev(k,10,j)= avele(k,10,j) + (avele(k,10,j) -
     &                                  elev(k,9,j))
   45  CONTINUE
c
c
c-----SECTION 3-----
c
c
      do 50 k=1,kno
            km = k-1
            ni = nlimk(k)
      do 50 n=1,ni
            nn=n+1
            avwid(k,n) = xwidth(k,1)+((xi(k,n)/x(k,ni))*(xwidth(k,2)
     &                  -xwidth(k,1)))
            avlen(k,n) = x(k,n) -x(k,n-1)
            if(k.gt.1) aslen(km,n) = (avwid(k,n)/2.)+(avwid(km,n)/2.)
            sumar(k,n) = avlen(k,n) * avwid(k,n)
      do 50 j=1,jno
            asd1 = (elev(k,n,j) - avele(k,n,j))/2.
            asd2 = (avele(k,n,j) - elev(k,n-1,j))/2.
            avdrop(k,n,j) = asd1 + asd2
            avslln(k,n,j) = sqrt((avdrop(k,n,j)**2) +
     &                                (avlen(k,n) **2))
            sinphi = avlen(k,n) / avslln(k,n,j)
            avvol(k,n,j) = (avdep(k,n,j)*avslln(k,n,j)*avwid(k,n)
     &                     * abs(sinphi)) * (1-stonec(j))
            avslp(k,n,j) = avdrop(k,n,j) / avlen(k,n)
            smln(k) = smln(k) + avlen(k,n)
       if(k.eq.1) goto 50
            asdrop(km,n,j) = avele(k,n,j) - avele(km,n,j)
            asslp(km,n,j)  = asdrop(km,n,1) / aslen(km,n)
            asslln(km,n,j) = sqrt(asdrop(km,n,j)**2 + aslen(km,n)**2)
   50 CONTINUE
C
C
C************SECTION 4
C
C     Now that most of the averaging proceedures have been removed, and
C     the remainder transferred to section 3 above, this section remains
C     simply to calculate the total area.
C
C
      sgarea = 0.
      do 70 k=1,kno
            ni=nlimk(k)
      do 60 n=1,ni
   60       area(k)=area(k)+sumar(k,n)
   70       sgarea=sgarea+area(k)
C
C
C************SECTION 5
C
C      The areal correction proceedure formerly housed in section 5
C      has been removed for the time being. Now that there is no need
C      to average between more than two sets of linepairs the areal
C      error should be small.
C      Only a few calculations remain although the best proceedure for
C      clevel has yet to be decided.
C
       sqk=sgarea/(1000.**2)
       sumkm=sumkm+sqkm
C
C
C
   80 CONTINUE
C
       return
       end
C
C''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

```
C
***************************************************************************
*
C
c SUBROUTINE    VCI
C
C
***************************************************************************
C
      function vci(UW)
c
c
c This function calculates the dynamic part of the
c soil strength algorithm.
c Created by J.E. Cochrane
c at the Department of Geography, University of Bristol, England.
c June, 1987.
c
c
c 'vci' is called by 'vouta'
c
c
      include 'spec.vcone'
      include 'spec.vcone2'
C
      uw=(uw*100/2.54)
      e3=(ua+uw)*tan(phib)
      s3=(e1-E2+E3)*E4
      sum=(s1*s2)-s3
      vci=sum
C
C
 2000 FORMAT(2X,' CI = ',F8.4)
      return
      end
```

```
c
c*******************************************************
c
c SUBROUTINE    VCNTL
c
c*******************************************************
c
      subroutine vcntl
c
c
c This subroutine controls the flow of control in the
c VSAS2 scheme.
c Created by J.E. Cochrane
c Original by Whitelaw, 1988.
c at the Geography Department, University of Bristol, England.
c December, 1987.
c
c
c 'vcntl' is called by 'link'
c
c 'vcntl' calls 'vmenu'
c              'bmsg1'
c              'bmsg2'
c              'vread1'
c              'vread2'
c              'vread3'
c              'vcone'
c              'vblkvol'
c              'vinter'
c              'vsurflo'
c              'vinit'
c              'vdrain'
c              'vouta3'
c              'voutb1'
c              'voutb2'
c              'voutb3'
c
c
      include 'spec.vgeom'
      include 'spec.vphys'
      include 'spec.vindex'
      include 'spec.vwater'
      include 'spec.vtime'
      include 'spec.vdata'
      include 'spec.vtitle'
      include 'spec.vcom2'
      include 'spec.vkgr'
      include 'spec.vcone'
      include 'spec.vcone2'
c
c
c
       dimension qqq(40)
       character fname*15,moi*15
       character answ*1
       logical fexist
c
c
```

```
      call vmenu
      if(nl.eq.999)goto 1000
   20 write(6,*)'INPUT NAME OF SEGMENT FILE'
      k1=1
      read(5,*,iostat=k1,err=20)fname
      inquire(file=fname,exist=fexist)
      if(.not.fexist)call bmsg1
      if(.not.fexist)goto 20
      open(9,file=fname,status='old',err=20)
      rewind 9
   22 write(6,*)'INPUT NAME OF STORM FILE'
      k1=1
      read(5,*,iostat=k1,err=22)fname
      inquire(file=fname,exist=fexist)
      if(.not.fexist)call bmsg1
      if(.not.fexist)goto 22
      open(3,file=fname,status='old',err=22)
      rewind 3
   24 write(6,*)'INPUT NAME OF INITIAL MOISTURE FILE'
      k1=1
      read(5,*,iostat=k1,err=24)fname
      inquire(file=fname,exist=fexist)
      if(.not.fexist)call bmsg1
      if(.not.fexist)goto 24
      open(4,file=fname,status='old',err=24)
      rewind 4
   26 write(6,*)'INPUT NAME OF SOIL DESCRIPTION FILE'
      k1=1
      read(5,*,iostat=k1,err=26)fname
      inquire(file=fname,exist=fexist)
      if(.not.fexist)call bmsg1
      if(.not.fexist)goto 26
      open(2,file=fname,status='old',err=26)
      rewind 2
      open(7,file='v3dt.out')
      rewind 7
c
        answ='y'
    5   write(6,*)'PLEASE ENTER NAME OF OUTPUT FILE'
        k1=1
        read(5,*,iostat=k1,err=5)moi
        inquire(file=moi,exist=fexist)
        if(fexist)call bmsg2(answ)
        if(answ.ne.'y')goto 5
        open(12,file=moi)
        SUMKM=0.
c
c'''''Read initial data and print headings
c
      call vread1
c     call voutal
        IEND=I-1
        M=1
c
c'''''Start of segment specific simulation. Read 2 contains
c'''''data for each segment.
c
  100 call vread2
```

```
      if(kno.ge.99)goto 340
      call vread3
      call vcone
        IMIN=0
        IHOUR=0
        IT=0
        IXM=0
C
C
      qq = 0.
c
c'''''Calculate the segment geometry and print it.
c
      call vblkvol
c     if(nouta.eq.0)call vouta2
c
c'''''Start of hourly simulation. IT represents hours, and DRAIN is
c'''''called every KREP times per hour, with LREP repetitions within
c'''''each call. The loop onto 161 allows for an increased timestep
c'''''to be used from the start of simulation until an 8. is found
c'''''in the data set, at which time the steps revert to 15 & 5 mins
c'''''or whatever the LREP & KREP values are set to below.
c
160     IT=IT+1
        IF(IT.GT.IEND) GOTO 100
        IF(PRECIP(IT).LT.8.)GOTO 161
        LREP=3
        KREP=4
        PRECIP(IT)=0.0
161     RAIN=PRECIP(IT)
c
c'''''INTER is called to calculate the effective precip and surflo
c'''''calcs the surface runoff. PNR is effective precipitation minus
c'''''the amount taken as surface runoff (ie so that it's not added
c'''''to the subsurface as well in drain)
c
        call vinter(IMONTH,IT,RAIN,PNET)
        CSS=0.
        IF (PNET.GT.0.)call vsurflo(A1,A2,PNET,CSS)
        PNR=((PNET*sgarea)-(PNET*(A1+A2)))/sgarea
C
c
c'''''The loop below calls INIT on the first pass for each segment
c'''''and DRAIN on every pass. The K passed to drain is not K as in
c'''''subsegments, but K from KREP
c
c
      DO 305 K=1,KREP
           if(it.eq.1.and.k.eq.1) call vinit
         call vdrain(K)
           IX=IXM+1
305        CONTINUE
C
c
c'''''The next bit (down to 310) adds together the direct runoff
c'''''(from SURFLO above) and the flow to the stream emerging
c'''''DRAIN. FLOW(K,J) is outflow from each depth (j) of each
c'''''subsegment (k).
```

```
c
      q=css/3.6
c
      do 310 k=1,kno
          eflow(k) = 0.
      DO 310 J=1,JNO
          eflow(k)=eflow(k) + flow(k,j)
  310     q=q+flow(k,j)
999   format('MAIN  :',f9.4)
c
C *** Accumulate all flows in m3/hr
c
          qq=qq+(q*3.6)
C
c
c'''''OUTA3 print an hourly results summary, OUTBland OUTB2 are
c'''''called to accumulate results to be printed at the end of
c'''''the simulation.
c
       if(ntmp.eq.0)call vouta3
      call voutbl(isgno,it)
C
      if (it.ge.(iend-1))qqq(isgno)=qq
C
c
      call voutb2(it,q,pnet,rain)

      goto 160
C *** New array to sum total runoff volumes for each segment
C
  340 call voutb3(it,mno)
C
        do 600 nnn=1,mno
  600   qqqq=qqqq+qqq(nnn)
        close(2)
        close(3)
        close(4)
        close(7)
        close(9)
        close(12)
C
        WRITE(7,500)
        WRITE(7,501)(NNN,QQQ(NNN),NNN=1,MNO)
        WRITE(7,502)QQQQ
500     FORMAT(1X,'OUTFLOW VOLUMES PER SEGMENT (M3)')
501     FORMAT(5X,I2,3X,F15.3)
502     FORMAT(1X,'TOTAL OUTFLOW FOR SIM. PERIOD (M3)',F15.3)
C
350     FORMAT('  END OF JOB')
C
 1000 return
c
      end
```

```
c
c*************************************************************************
**
c
c FUNCTION    VCON
c
c*************************************************************************
**
c
      function vcon(k,n,j,tpbv)
c
c
c This function calculates hydraulic conductivity according
c to Cambell's method.
c Created by Bernier,1982; Whitelaw, 1988.
c
c
c 'vcon' is called by 'vdrain'
c
      include 'spec.vphys'
      include 'spec.vindex'
c
c
      cbc = cb(k,n,j)
      c2bp2=(cbc*2.)+2.
      vcon = cmax(k,n,j)*((tpbv/poros(k,n,j))**c2bp2)
c
c
      return
      end
c
```

```
c
**************************************************************************
c
c SUBROUTINE    VCONE
c
c
*************************************************************************
c
      subroutine vcone
c
c This subroutine calculates the static part of the
c soil strength algorithm.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c
c
c 'vcone' is called by 'vcntl'
c
c
      include 'spec.vcone'
      include 'spec.vcone2'
c
      real m
      alpha=(22*2*alpha)/(7*360)
      phi=(22*2*phi)/(7*360)
      phib=(22*2*phib)/(7*360)
      m=(4*sin(phi))/(3*(1+sin(phi)))
 1000 FORMAT(2X,'M =  ',F9.7)
      b1=(tan(alpha)+tan(phi))*tan(alpha)*8
      a1=1-m
      a2=2-m
      a3=3-m
      b2=d*d*a1*a2*a3
      b3=(gg*(1/tan(phi)))**m
      b4=(gamma)**a1
      s1=(b1*b3*b4)/b2
c
      d1=((c/gamma)*(1/tan(phi))+aal+zz)**a3
      d2=((c/gamma)*(1/tan(phi))+zz)**a3
      d3=((c/gamma)*(1/tan(phi))+zz)**a2
      d3=d3*a3*aal
      s2=d1-d2-d3
c
      e1=c*tan(alpha)*(1/tan(phi))
      e2=ua*tan(phi)
      e4=(4*aal*aal*tan(alpha))/(d*d)
c
  876 CONTINUE
      return
      end
```

```
c
c*********************************************************
c
c SUBROUTINE     VDRAIN
c
c*********************************************************
c
      subroutine vdrain(kk)
c
c     Much of the code here is taken directly from the old drain
c     subroutine in vsas2. It has been modified to include the
c     new array sizes and variable names for vsas3 and to call
c     latflo to transfer moisture between the numerical grids
c     A k loop is introduced to allow simulation of each grid in
c     turn.
c Original by Bernier, 1982;Whitelaw, 1988.
c
c
c 'vdrain' is called by 'vcntl'
c
c 'vdrain' calls 'vlatflo2'
c               'vxmatrx'
c
c
c
c
      include 'spec.vgeom'
      include 'spec.vphys'
      include 'spec.vindex'
      include 'spec.vdata'
      include 'spec.vwater'
      include 'spec.vkgr'
c
      dimension xflo(5,16,5)
c
c
c     fac converts cm/hr to m/hr and divides the xflos into fractions
c     of hours depending on the iteration controls lrep & krep
c
      fac = 0.01 / (float(krep*lrep))
c
      do 100 l = 1,lrep
            ix = 2
      do  95 k = 1,1
            ni = nlimk(k)
   30 do 90 jss= 1,jno
            js = jno-jss+1
            jx = js+(2-ix)
            if(jx.gt.jno)  goto 90
      do 90 ns = 1,ni
            nx = ns-(ix-1)
c
c     The only iteration controls currently engaged are to omit
c     the krep/lrep iterations (usually 5 mins) if the moisture
c     potential is less than -3.0 cm. An input variable or
c     internally determined value might replace this situation.
c
c      if (l.ne.1.and.(poten(k,nx,jx).lt.(-3.).and.nx.ne.0))
```

```
c     & goto 70
      if(ix.eq.1) goto 45
c
c-----section 2-----adds net precip to surface and calculates
c                   conductivities and matric potentials.
c
      tpbv = smcvol(k,ns,js)/avvol(k,ns,js)
      if (js.eq.1) smcvol(k,ns,js)=smcvol(k,ns,js)+(pnr*sumar(k,ns)
     &                             /(100.*lrep*krep))
      tpbv = (tpbv+smcvol(k,ns,js)/avvol(k,ns,js))/2.
c      pres(k,ns,js)=amax1(pres(k,n,js),(clevel-avele(k,ns,js))
c     &                             *100.)
      tpbv = amin1(poros(k,ns,js),tpbv)
      poten(k,ns,js) = vxmatrx(k,ns,js,tpbv)
c      if (pres(k,ns,js).gt.0.) poten(k,ns,js) = pres(k,ns,js)
      conduc(k,ns,js) =   vcon(k,ns,js,tpbv)
   45 if (avvol(k,ns,js).le.0.)      goto 90
      if (tpbv.lt.0.01)              goto 90
      if (nx.ne.0)                   goto 50
c
c-----section 3-----flow from streamside elements into stream
c
      grad = amax1(0.,(((avele(k,ns,js)-(avele(k,ns,5))+clevel))
     &       *100. + poten(k,ns,js))/dist(k,ns,js,2))
      grad = 0.0
      uscon = conduc(k,ns,js)
      goto 65
c
c-----section 4-----calcs parameters for flow between elements
c
  50  eldrop = avele(k,ns,js)-avele(k,nx,jx)
      potdif = poten(k,ns,js) - poten(k,nx,jx)
      grad = (potdif + (eldrop * 100.)) / dist(k,ns,js,ix)
      if (ix.eq.1) then
                   top = (avdep(k,ns,js)+avdep(k,nx,jx))*100.
                  bots = (avdep(k,ns,js)*100.)/conduc(k,ns,js)
                  botx = (avdep(k,nx,jx)*100.)/conduc(k,nx,jx)
                 uscon = top / (bots+botx)
      else
                   top = (avslln(k,ns,js)+avslln(k,nx,jx))*100.
                  bots = (avslln(k,ns,js)*100.)/conduc(k,ns,js)
                  botx = (avslln(k,nx,jx)*100.)/conduc(k,nx,jx)
                 uscon = top / (bots+botx)
      endif
c
c-----section 5-----calculates flows and updates moisture contents
c
   65 xflo(k,ns,js) = grad * sarea(k,ns,js,ix) * uscon * fac
      if(ix.eq.2) xflo(k,ns,js) = xflo(k,ns,js) * shift(k)
      if (nx.eq.0) goto 85
      if (ix.eq.2) goto 80
      extra = smcvol(k,nx,jx) + xflo(k,ns,js) - smcmax(k,nx,jx)
      if(extra.le.0.) goto 80
c
      xflo(k,ns,js)  = xflo(k,ns,js) - extra
c
   80 smcvol(k,nx,jx)   = smcvol(k,nx,jx)    + xflo(k,ns,js)
   85 smcvol(k,ns,js)   = smcvol(k,ns,js)    - xflo(k,ns,js)
```

```
c
   90 CONTINUE
c
c
c
      if (ix.eq.1) then
                       ix = 2
                       goto 95
                   else
                        if(k.lt.kno) call vlatflo2(k)
                       ix=1
                   endif
c
c
c      xflo has now been declared as an array in order to allow the
c      outflows to be accumulated after the call to latflo has been
c      made.
c
       if(1.ne.1.or.kk.ne.1) goto 117
       do 115 j = 1,jno
  115         flow(k,j) = 0.
  117  CONTINUE
c
       do 120 j = 1,jno
              flow(k,j) = flow(k,j) + ( xflo(k,1,j) / 3.6 )
  120 CONTINUE
c
c
      goto 30
c
   95 CONTINUE
  100 CONTINUE
c
c-----section 6-----slopewise flows from surface layers
c
      do 110 k = 1,1
            ni = nlimk(k)
      do 110 n = 1,ni
                 extra = amax1(0.,smcvol(k,n,1)-smcmax(k,n,1))
                 if (n.eq.1) flow(k,1) = flow(k,1) + extra
                 smcvol(k,n,1) = smcvol(k,n,1)      - extra
                 if (n.gt.1) smcvol(k,n-1,1)=smcvol(k,n-1,1)+extra
      do 110 j = 1,jno
                 pres(k,n,j) = amax1(0.,pres(k,n,j))
                 pbv(k,n,j)  = smcvol(k,n,j) / avvol(k,n,j)
  110 CONTINUE
c
c
c
      return
      end
c
c
c--------------------------------------------------------------
```

```
c
c*********************************************************
c
c SUBROUTINE     VINIT
c
c*********************************************************
c
      subroutine vinit
c
c Original by Bernier, 1982;Whitelaw, 1988.
c
c
c 'vinit' is called by 'vcntl'
c
c
      include 'spec.vgeom'
      include 'spec.vphys'
      include 'spec.vindex'
      include 'spec.vwater'
      include 'spec.vdata'
      include 'spec.vkgr'
      include 'spec.vcom2'
c
C**** This is a revised version of the INIT subroutine designed
C     to allow the use of three dimensional proceedures in DRAIN
C     Moisture contents and soil properties are still allocated
C     in the same manner, with an outside loop for k (kno-1)
C     being the number of internal grid networks. The geometric
C     properties are derived from the new version of BLKVOL .
C
C
      do 210 k=1,kno
             comlen = 0.
             nii = nlimk(k)
      do 210 n=1,nii
C
C     This section allocates soil moisture properties to the
C     geometric elements from BLKVOL.
C
c
c'''''The three proportional distances represented by POS 1-3
c'''''are converted to real distances and used to choose which of
c'''''the soil property set should be given to that element.
c'''''It should be noted that the distances are not the same for
c'''''each subsegment
c
          tpos1 = ((horiz(k,1)+horiz(k+1,1))/2.)*pos(1)
          tpos2 = ((horiz(k,1)+horiz(k+1,1))/2.)*pos(2)
          comlen = comlen + avlen(k,n)
c
c
             if(comlen.gt.tpos2) then
                 do 250 j=1,jno
                   ca(k,n,j)=ca3(k,j)
                   cb(k,n,j)=cb3(k,j)
                   cmax(k,n,j)=cmax3(k,j)
                   poros(k,n,j)=poros3(k,j)
  250            CONTINUE
```

```
            elseif (comlen.gt.tpos1.and.comlen.le.tpos2) then
                do 260 j=1,jno
                  ca(k,n,j)=ca2(k,j)
                  cb(k,n,j)=cb2(k,j)
                  cmax(k,n,j)=cmax2(k,j)
                  poros(k,n,j)=poros2(k,j)
  260           CONTINUE
            else
                do 270 j=1,jno
                  ca(k,n,j)=ca1(k,j)
                  cb(k,n,j)=cb1(k,j)
                  cmax(k,n,j)=cmax1(k,j)
                  poros(k,n,j)=poros1(k,j)
  270           CONTINUE
            endif
  210 CONTINUE
C
C     The next section calcultes areas and distances for inter element
C     flows. The recalculation is required here to allow for possible
C     future uses of RECOM
C
      do 300 k=1,kno
            nii = nlimk(k)
      do 300 n=1,nii
      do 280 j = 1,jno
      if(j.ne.jno)dist(k,n,j,1)=(avdep(k,n,j)+avdep(k,n,j+1))*50.
      if(n.ne.1)dist(k,n,j,2)=(avslln(k,n,j)+avslln(k,n-1,j))*50.
      if(n.eq.1)dist(k,n,j,2)=avslln(k,n,j)*100.
      if(k.lt.kno)  dist(k,n,j,3)=(asslln(k,n,j)+asslln(k+1,n,j))*50.
C
      if(j.ne.jno) sarea(k,n,j,1)=avlen(k,n)*avwid(k,n)*(1-stonec(j)
     *                                    +stonec(j+1))/2.
                   sarea(k,n,j,2)=(1-stonec(j))*avvol(k,n,j)/avslln
     *                                 (k,n,j)*avlen(k,n)/avslln(k,n,j)
                   sarea(k,n,j,3)=(1-stonec(j))*avvol(k,n,j)/avwid(k,n)
                   smcmax(k,n,j)=poros(k,n,j)*avvol(k,n,j)
C
C
C
C
      xlevel(j)=amin1(xlevel(j),amax1(clevel,avele(k,1,j)
     &                                     -avdrop(k,1,j)/2.))
            pbv(k,n,j) = pbv(k,n,j) * poros(k,n,j)
            pbv(k,n,j) = amin1(poros(k,n,j),pbv(k,n,j))
            smcvol(k,n,j) = pbv(k,n,j) * avvol(k,n,j)
  280 CONTINUE
C
  300 CONTINUE
C
C
C
      return
      end
C
C
```

```
c
c************************************************************
c
c SUBROUTINE    VINTER
c
c************************************************************
c
      SUBROUTINE vinter(MONTH,IT,PRECIP,PNET)
C
C     The purpose of this subroutine is to collect rainfall in a
C     storage represented by AIMAX until it is filled and then to
C     transmit any further rainfall directly to the catchment.
C     Every hour the storage yields 20% of its volume to the
C     catchment as stemflow and drip. Storage values vary with
C     season as represented by the month. A rain free period of
C     24 hours will reset the storage to 0.
c Original by Bernier, 1982;Whitelaw, 1988.
c
c
c 'vinter' is called by 'vcntl'
C
C
C     0.249cm (dormant)   0.254cm (growing)
C
      AIMAX=0.249
      IF(MONTH .GE.5.AND.MONTH.LT.11)AIMAX=0.254
      IF(IT.EQ.1)AIDEF=AIMAX
      IF(IT.EQ.1)ICOUNT=IT
      IF(PRECIP.GT.0)ICOUNT=IT
      IF(IT-ICOUNT.GE.24)AIDEF=AIMAX
      IF(IT-ICOUNT.GE.24)ICOUNT=IT
      XLOSS=AMIN1(PRECIP,AIDEF)
      AIDEF=AIDEF-XLOSS
      PNET=(PRECIP-XLOSS)+((AIMAX-AIDEF)*05./100.)
      AIDEF=AIDEF+((AIMAX-AIDEF)*20./100.)
C
      RETURN
      END
```

```
c
c*******************************************************
c
c SUBROUTINE    VLATFLO
c
c*******************************************************
c
c
      subroutine vlatflo(k)
c
c
c     This is a service subroutine for drain to calculate
c     the lateral flows between the grids within segments.
c     It is called after the stream directed slopewise flows
c     have been done in drain (ie when ix = 2)
c
c
      include 'spec.vgeom'
      include 'spec.vphys'
      include 'spec.vindex'
      include 'spec.vkgr'
c
c
c
      fac=1./(float(krep*lrep))
      ni = min0 (nlimk(k),nlimk(k+1))
c
      do 10 j = 1,jno
      do 20 n = 1,ni
c
c
c
c
            eldrop = avele(k,n,j) - avele(k+1,n,j)
            grad  =(poten(k,n,j)-poten(k+1,n,j)+eldrop
     &                          *100.)/dist(k,n,j,3)
            tcon  = amax1(conduc(k+1,n,j),amin1(conduc(k,n,j)
     &              ,cmax(k+1,n,j)))
            uscon = (dist(k,n,j,3)/50.)/((asslln(k,n,j)/
     &              conduc(k,n,j))+(asslln(k+1,n,j)/tcon))
            xflo  = grad*sarea(k+1,n,j,3)*uscon*fac
c
c
            smcvol(k+1,n,j) = smcvol(k+1,n,j) + xflo
            smcvol(k,n,j)   = smcvol(k,n,j)   - xflo
c
c
   20 CONTINUE
   10 CONTINUE
c
c
      return
      end
```

```
c
c*************************************************************
c
c SUBROUTINE     VLATFLO2
c
c*************************************************************
c
c
      subroutine vlatflo2(k)
c
c
c     This is a service subroutine for drain to calculate
c     the lateral flows between the grids within segments.
c     It is called after the stream directed slopewise flows
c     have been done in drain (ie when ix = 2)
c Original by Bernier, 1982;Whitelaw, 1988.
c
c
c 'vlatflo2' is called by 'vdrain'
c
c
      include 'spec.vgeom'
      include 'spec.vphys'
      include 'spec.vindex'
      include 'spec.vcom2'
      include 'spec.vkgr'
c
c
c
      fac= 0.01 /(float(krep*lrep))
      ni = min0 (nlimk(k),nlimk(k+1))
c
      do 10 j = 1,jno
      do 20 n = 1,ni
c
c
      recpt = xi(k,n) - roffst(k+1)
      if(recpt.le.0.) goto 20
      if(recpt.gt.x(k,ni)) goto 20
c
      do 30 n2=1,ni
             xpt1 = xi(k+1,n2)
             xpt2 = xi(k+1,n2+1)
            if(xpt1.le.recpt.and.xpt2.gt.recpt)
     &      nkeypt = n2
   30 CONTINUE
c
      test1 = smcvol(k,n,j)/avvol(k,n,j)
      test2 = smcvol(k+1,nkeypt,j)/avvol(k+1,nkeypt,j)
      test3 = smcvol(k+1,nkeypt+1,j)/avvol(k+1,nkeypt+1,j)
      if(test1.le.0.10.or.test2.le.0.10.or.test3.le.0.10)
     &  goto 20
c
      rlen2 = xi(k+1,nkeypt+1) - recpt
      rlen1 = recpt - xi(k+1,nkeypt)
      if(rlen1.lt.0..or.rlen2.lt.0.)goto 20
c
      weigh1 = rlen2/(rlen1+rlen2)
      weigh2 = rlen1/(rlen1+rlen2)
c
      fdist = (assl1n(k,n,j)+((assl1n(k+1,nkeypt,j)*weigh1)+
     &       (assl1n(k+1,nkeypt+1,j)*weigh2))) / 2.
      eldrop = avele(k,n,j)-((avele(k+1,nkeypt,j)*weigh1)+
     &         (avele(k+1,nkeypt+1,j)*weigh2))
      potdif = poten(k,n,j)-((poten(k+1,nkeypt,j)*weigh1)+
     *         (poten(k+1,nkeypt+1,j)*weigh2))
      if (eldrop.le.0.) potdif = abs(potdif)
      grad = (potdif + (eldrop * 100.)) /(fdist*100.)
c
c
        top = (assl1n(k,n,j)+assl1n(k+1,nkeypt,j))*100.
       bots = (assl1n(k,n,j)*100.)/conduc(k,n,j)
       botx = (assl1n(k+1,nkeypt,j)*100.)/conduc(k+1,nkeypt,j)
      uscon1 = top / (bots+botx)
c
        top = (assl1n(k,n,j)+assl1n(k+1,nkeypt+1,j))*100.
       botx = (assl1n(k+1,nkeypt+1,j)*100.)/conduc(k+1,nkeypt+1,j)
      uscon2 = top / (bots+botx)
c
      uscon = (uscon1*weigh1) + (uscon2*weigh2)
c
      farea1 = sarea(k+1,nkeypt,j,3) * weigh2
      farea2 = sarea(k+1,nkeypt+1,j,3) * weigh1
c
      farea = farea1 + farea2
c
      xflo = grad * farea * uscon * fac
c
      xflo1 = xflo * weigh1
      xflo2 = xflo * weigh2
c
c
c
c
            smcvol(k+1,nkeypt,j) = smcvol(k+1,nkeypt,j) + xflo1
            smcvol(k+1,nkeypt+1,j) =smcvol(k+1,nkeypt+1,j)
     &                                                   + xflo2
            smcvol(k,n,j) = smcvol(k,n,j) - (xflo1+xflo2)
c
c
   20 CONTINUE
   10 CONTINUE
c
c
      return
      end
```

```
c
c*******************************************************
c
c SUBROUTINE      VMENU
c
c*******************************************************
c
       subroutine vmenu
c
c
c This subroutine allows the user to interactively edit
c input files for VSAS2 or run VSAS2.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c December, 1987.
c
c
c 'vmenu' is called by 'vcntl'
c
c 'vmenu' calls 'bmsg2'
c               'vview'
c               'bmsg1'
c               'vread1'
c               'vread2'
c               'vread3'
c
c
      include 'spec.vgeom'
      include 'spec.vphys'
      include 'spec.vindex'
      include 'spec.vwater'
      include 'spec.vtime'
      include 'spec.vdata'
      include 'spec.vcom2'
      include 'spec.vkgr'
      include 'spec.vcone'
c
c
      character fname*20
      character answ*1
      logical fexist
c
c
    1 write(6,*)'MENU SELECTION'
      write(6,*)'1. CREATE A NEW DATA FILE'
      write(6,*)'2. CHANGE AN EXISTING DATA FILE'
      write(6,*)'3. RUN VSAS2'
      write(6,*)'4. EXIT WITHOUT A RUN'
      read(5,*)ii
      goto(5,430,1000,1010)ii
    5 k1=1
   10 write(6,*)'INPUT NUMBER OF SEGMENTS'
      write(6,*)'PRESENT VALUE IS ',mno
      read(5,*,iostat=k1,err=10)mno
      if(jr.gt.o)goto 435
   40 write(6,*)'INPUT NAME OF SEGMENT'
      write(6,*)'PRESENT NAME IS ',name
      k1=1
      read(5,*,iostat=k1,err=40)name
      if(jr.gt.0)goto 435
   45 write(6,*)'INPUT ANY NOTE YOU WISH TO HAVE IN DATA FILE'
      write(6,*)'PRESENT NOTE IS ',note
      k1=1
      read(5,*,iostat=k1,err=45)note
      if(jr.gt.0)goto 435
   50 write(6,*)'INPUT NO. OF COARSE SCALE ITERATIONS IN 1 HOUR'
      write(6,*)'PRESENT VALUE IS ',krep
      k1=1
      read(5,*,iostat=k1,err=50)krep
      if(jr.gt.0)goto 435
   55 write(6,*)'INPUT NO. OF FINE SCALE ITERATIONS WITHIN COARSE SCALE'
      write(6,*)'PRESENT VALUE IS ',lrep
      k1=1
      read(5,*,iostat=k1,err=55)lrep
      if(jr.gt.0)goto 435
   70 write(6,*)'INPUT NO. OF HOURS BETWEEN PRINTOUTS'
      write(6,*)'PRESENT VALUE IS ',npo
      k1=1
      read(5,*,iostat=k1,err=70)npo
      if(jr.gt.0)goto 435
   71 write(6,*)'INPUT OPTION FOR PRINTING SEGMENT GEOMETRY'
      write(6,*)'0 = PRINT,    >0 = NO PRINT'
      write(6,*)'PRESENT VALUE IS ',nouta
      k1=1
      read(5,*,iostat=k1,err=71)nouta
      if(jr.gt.0)goto 435
c

c
c
  120 write(6,*)'INPUT OPTION FOR CALC. AREAS OF 100 & 75% SATURATION'
      write(6,*)'0 = CALCULATED,   >0 = NOT CALCULATED'
      write(6,*)'PRESENT VALUE IS ',ntmp
      k1=1
      read(5,*,iostat=k1,err=120)ntmp
      if(jr.gt.0)goto 435
  125 write(6,*)'INPUT SEGMENT NUMBER'
      write(6,*)'PRESENT VALUE IS ',isgno
      k1=1
      read(5,*,iostat=k1,err=125)isgno
      if(jr.gt.0)goto 435
  130 write(6,*)'INPUT TOTAL NUMBER OF SEGMENTS'
      write(6,*)'PRESENT VALUE IS ',kno
      k1=1
      read(5,*,iostat=k1,err=130)kno
      if(jr.gt.0)goto 435
  140 write(6,*)'INPUT NUMBER OS SOIL LAYERS'
      write(6,*)'PRESENT VALUE IS ',jno
      k1=1
      read(5,*,iostat=k1,err=140)jno
      if(jr.gt.0)goto 435
  145 write(6,*)'INPUT ROUTING DELAY (MINS)'
      write(6,*)'PRESENT VALUE IS ',irout
      k1=1
      read(5,*,iostat=k1,err=145)irout
      if(jr.gt.0)goto 435
```

```
 150 write(6,*)'INPUT SEGMENT AREA'
     write(6,*)'PRESENT VALUE IS ',sgarea
     k1=1
     read(5,*,iostat=k1,err=150)sgarea
     if(jr.gt.0)goto 435
 160 write(6,*)'INPUT CLEVEL'
     write(6,*)'PRESENT VALUE IS ',clevel
     k1=1
     read(5,*,iostat=k1,err=160)clevel
     if(jr.gt.0)goto 435
c
 198 do 200 i=1,jno
 202 write(6,*)'INPUT STONE CONTENT IN LAYER ',jno
     write(6,*)'PRESENT VALUE IS ',stonec(i)
     k1=1
 200 read(5,*,iostat=k1,err=198)stonec(i)
     if(jr.gt.0)goto 435
 215 do 230 i=1,1
 222 write(6,*)'INPUT NUMBER OF INCREMENTS ON SLOPE'
     write(6,*)'PRESENT VALUE IS ',nlimk(i)
     k1=1
 230 read(5,*,iostat=k1,err=222)nlimk(i)
     if(jr.gt.0)goto 435
 231 do 240 k=1,kno
     nn=nlimk(kno)
     do 238 n=1,nn
 235 write(6,*)'INPUT ELEVATION NUMBER ',n,' FOR SEGMENT NO. ',k
     write(6,*)'PRESENT VALUE IS ',yi(k,n)
     k1=1
     read(5,*,iostat=k1,err=235)yi(k,n)
     if(jr.gt.0)goto 435
 236 write(6,*)'INPUT SOIL THICKNESS OF INCREMENT ',n
     write(6,*)'PRESENT VALUE IS ',depmax(k,n)
     k1=1
     read(5,*,iostat=k1,err=236)depmax(k,n)
     if(jr.gt.0)goto 435
 237 write(6,*)'INPUT DISTANCE FROM STREAM OF INCREMENT ',n,' SEGMENT ',k
     write(6,*)'PRESENT VALUE IS ',xi(k,n)
     k1=1
     read(5,*,iostat=k1,err=237)xi(k,n)
     if(jr.gt.0)goto 435
 238 CONTINUE
 240 CONTINUE
     if(jr.gt.0)goto 435
 241 do 250 i=1,kno
 242 write(6,*)'INPUT STREAM FRONT WIDTH OF SEGMENT NO. ',i

     write(6,*)'PRESENT VALUE IS ',xwidth(i,1)
     k1=1
     read(5,*,iostat=k1,err=242)xwidth(i,1)
 245 write(6,*)'INPUT WATERSHED WIDTH ON SEG NO. ',i
     write(6,*)'PRESENT VALUE IS ',xwidth(i,2)
     k1=1
 250 read(5,*,iostat=k1,err=245)xwidth(i,2)
     if(jr.gt.0)goto 435
 251 do 257 i=1,kno
 252 write(6,*)'INPUT LEFT SIDE DISTANCE ON SEG NO. ',i
     write(6,*)'PRESENT VALUE IS ',horiz(i,1)
```

```
      k1=1
      read(5,*,iostat=k1,err=252)horiz(i,1)
  255 write(6,*)'INPUT RIGHT HAND SIDE DISTANCE ON SEG NO. ',i
      write(6,*)'PRESENT VALUE IS ',horiz(i,2)
      k1=1
  257 read(5,*,iostat=k1,err=255)horiz(i,2)
      if(jr.gt.0)goto 435
  258 write(6,*)'INPUT DEPTH OF TOP LAYER'
      write(6,*)'PRESENT VALUE IS ',adepth
      k1=1
      read(5,*,iostat=k1,err=258)adepth
      if(jr.gt.0)goto 435
  260 do 264 i=2,jno-1
  261 write(6,*)'INPUT PROPORTION OF DEPTH (MINUS TOP LAYER) FOR LAYER ',i
      k1=1
      write(6,*)'PRESENT VALUE IS ',bdepth(i)
      k1=1
  264 read(5,*,iostat=k1,err=261)bdepth(i)
      if(jr.gt.0)goto 435
c
  265 write(6,*)'INPUT ATEST'
      write(6,*)'PRESENT VALUE IS ',atest
      k1=1
      read(5,*,iostat=k1,err=265)atest
      if(jr.gt.0)goto 435
  266 write(6,*)'INPUT CDEPTH'
      write(6,*)'PRESENT VALUE IS ',cdepth
      k1=1
      read(5,*,iostat=k1,err=266)cdepth
      if(jr.gt.0)goto 435
  267 write(6,*)'INPUT A1'
      write(6,*)'PRESENT VALUE IS ',a1
      k1=1
      read(5,*,iostat=k1,err=267)a1
      if(jr.gt.0)goto 435
  270 write(6,*)'INPUT A2'
      write(6,*)'PRESENT VALUE IS ',a2
      k1=1
      read(5,*,iostat=k1,err=270)a2
      if(jr.gt.0)goto 435
      do 275 i=1,kno-1
  274 write(6,*)'INPUT ROFFST NUMBER ',i
      write(6,*)'PRESENT VALUE IS ',roffst(i)
      k1=1
  275 read(5,*,iostat=k1,err=274)roffst(i)
      if(jr.gt.0)goto 435
c
  276 do 300 k=1,kno
  277 do 283 j=1,jno
  278 write(6,*)'INPUT CMAX1 FOR LAYER ',j
      write(6,*)'PRESENT VALUE IS ',cmax1(k,j)
      k1=1
      read(5,*,iostat=k1,err=277)cmax1(k,j)
  279 write(6,*)'INPUT POROS1 FOR LAYER ',j
      write(6,*)'PRESENT VALUE IS ',poros1(k,j)
      k1=1
  280 read(5,*,iostat=k1,err=278)poros1(k,j)
  281 write(6,*)'INPUT CA1 FOR LAYER ',j
```

```
      write(6,*)'PRESENT VALUE IS ',ca1(k,j)
      k1=1
      read(5,*,iostat=k1,err=281)ca1(k,j)
  282 write(6,*)'INPUT CB1 FOR LAYER ',j
      write(6,*)'PRESENT VALUE IS ',cb1(k,j)
      k1=1
      read(5,*,iostat=k1,err=282)cb1(k,j)
      if(jr.gt.0)goto 435
  283 CONTINUE
  292 write(6,*)'INPUT MEAN LENGTH OF SEGMENT  ',k
      write(6,*)'PRESENT VALUE IS ',hlen(k)
      k1=1
      read(5,*,iostat=k1,err=292)hlen(k)
      if(jr.gt.0)goto 435
  300 CONTINUE
      do 304 k=1,3
  302 write(6,*)'INPUT POS NO. ',k
      write(6,*)'PRESENT VALUE IS ',pos(k)
      k1=1
      read(5,*,iostat=k1,err=302)pos(k)
  304 CONTINUE
      if(jr.gt.0)goto 435
  400 write(6,*)'INPUT INTERNAL ANGLE OF FRICTION PHI'
      write(6,*)'PRESENT VALUE IS ',phi
      k1=1
      read(5,*,iostat=k1,err=400)phi
      write(6,*)'INPUT REMOULDED ANGLE OF FRICTION  PHIB'
      write(6,*)'PRESENT VALUE IS ',phib
      k1=1
      read(5,*,iostat=k1,err=400)phib
      if(jr.gt.0)goto 435
  405 write(6,*)'INPUT SOIL COHESION'
      write(6,*)'PRESENT VALUE IS ',c
      k1=1
      read(5,*,iostat=k1,err=405)c
      if(jr.gt.0)goto 435
  410 write(6,*)'INPUT SOIL SHEAR MODULUS'
      write(6,*)'PRESENT VALUE IS ',gg
      k1=1
      read(5,*,iostat=k1,err=410)gg
      if(jr.gt.0)goto 435
  420 write(6,*)'INPUT DEPTH OF CONE TIP'
      write(6,*)'PRESENT VALUE IS ',zz
      k1=1
      read(5,*,iostat=k1,err=420)zz
      if(jr.gt.0)goto 435
c
  285 write(6,*)'INPUT NUMBER OF HOURS OF RAIN '
      write(6,*)'PRESENT VALUE IS ',ir
      k1=1
      read(5,*,iostat=k1,err=285)ir
      if(jr.gt.0)goto 435
  286 do 290 i=1,ir
  287 write(6,*)'INPUT PRECIPITATION (m) FOR HOUR ',i
      write(6,*)'PRESENT VALUE IS ',precip(i)
      k1=1
  290 read(5,*,iostat=k1,err=287)precip(i)
      if(jr.gt.0)goto 435
```

```
c
  320 write(6,*)'INPUT NAME OF NEW SEGMENT FILE'
      k1=1
      answ='y'
      read(5,*,iostat=k1,err=320)fname
      inquire(file=fname,exist=fexist)
      if(fexist)call bmsg2(answ)
      if(answ.ne.'y')goto 320
      open(9,file=fname,err=320)
      rewind 9
  321 write(6,*)'INPUT NAME OF NEW STORM FILE'
      answ='y'
      k1=1
      read(5,*,iostat=k1,err=321)fname
      inquire(file=fname,exist=fexist)
      if(fexist)call bmsg2(answ)
      if(answ.ne.'y')goto 321
      open(3,file=fname,err=321)
      rewind 3
  322 write(6,*)'INPUT NAME OF NEW INITIAL MOISTURE FILE'
      k1=1
      answ='y'
      read(5,*,iostat=k1,err=322)fname
      inquire(file=fname,exist=fexist)
      if(fexist)call bmsg2(answ)
      if(answ.ne.'y')goto 322
      open(4,file=fname,err=322)
      rewind 4
  323 write(6,*)'INPUT NAME OF NEW SOIL DESCRIPTION FILE'
      k1=1
      answ='y'
      read(5,*,iostat=k1,err=323)fname
      inquire(file=fname,exist=fexist)
      if(fexist)call bmsg2(answ)
      if(answ.ne.'y')goto 323
      open(2,file=fname,err=323)
      rewind 2
  324 call vview
      close (9)
      close (3)
      close (4)
      close (2)
  325 write(6,*)'INPUT ANY NUMBER TO CONTINUE'
      read(5,*,iostat=k1,err=325)icheck
      goto 1
c
c
c THIS SECTION ALLOWS YOU TO CHANGE A PARAMETER VALUE
c
c
  430 write(6,*)'INPUT NAME OF SEGMENT FILE TO BE READ'
      k1=1
      read(5,*,iostat=k1,err=430)fname
      inquire(file=fname,exist=fexist)
      if(.not.fexist)call bmsg1
      if(.not.fexist)goto 430
      open(9,file=fname)
      rewind 9
  431 write(6,*)'INPUT STORM FILE NAME'
      k1=1
      read(5,*,iostat=k1,err=431)fname
      inquire(file=fname,exist=fexist)
      if(.not.fexist)call bmsg1
      if(.not.fexist)goto 431
      open(3,file=fname)
      rewind 3
  432 write(6,*)'INPUT INITIAL MOISTURE FILE NAME'
      k1=1
      read(5,*,iostat=k1,err=432)fname
      inquire(file=fname,exist=fexist)
      if(.not.fexist)call bmsg1
      if(.not.fexist)goto 432
      open(4,file=fname)
      rewind 4
  433 write(6,*)'INPUT SOIL DESCRIPTION FILE NAME'
      k1=1
      read(5,*,iostat=k1,err=433)fname
      inquire(file=fname,exist=fexist)
      if(.not.fexist)call bmsg1
      if(.not.fexist)goto 433
      open(2,file=fname)
      rewind 2
      call vread1
      call vread2
      call vread3
      close(9)
      close(3)
      close(4)
      close(2)
      iset=1
      k=6
  435 write(6,*)'PLEASE SELECT THE IDENTIFYING VALUE OF THE PARAMETER'
      write(6,*)'YOU WISH TO CHANGE FROM THE FOLLOWING SERIES OF LISTS'
      write(6,*)'   '
      write(6,*)'1.METEOROLOGICAL DATA'
      write(6,*)'2.PRECIPITATION DATA'
      write(6,*)'3.SOILS DATA'
      write(6,*)'4.RUN DEFINITION DATA'
      write(6,*)'5.SAVE FILE'
      write(6,*)'6.EXIT TO MAIN MENU'
      k1=1
      read(5,*,iostat=k1,err=435)kr
      goto(501,502,503,504,320,1)kr
      if(kr.gt.4)goto 1000
c
  501 write(6,*)'CHOOSE FROM THE FOLLOWING PARAMETERS'
c
      write(6,*)'1.PRECIPITATION VALUES'
      write(6,*)'2. COARSE ITERATION TIME INTERVAL'
      write(6,*)'3. FINE ITERATION TIME INTERVAL'
      write(6,*)'4. OPTION FOR PRINT OF SEGMENT GEOMETRY'
      write(6,*)'5. OPTION FOR CALC. OF 100% AND 75% SATURATED AREAS'
      write(6,*)'6.STONE CONTENT OF EACH LAYER'
      write(6,*)'7. SEGMENT NUMBER'
      write(6,*)'8. INCREMENTS ON SLOPE'
      write(6,*)'9.NUMBER OF SEGMENTS'
      write(6,*)'10. NUMBER OF SOIL LAYERS'
      write(6,*)'11. SEGMENT DIMENSIONS'
      write(6,*)'12. ROUTING DELAY'
      write(6,*)'13. SEGMENT DISTANCES'
      write(6,*)'14.***EXIT***'
c
c
      k1=1
      read(5,*,iostat=k1,err=501)jr
      goto(502,50,55,71,120,198,125,215,130,
     & 140,241,145,251,435)jr
       goto 1
c
  502 write(6,*)'CHOOSE FROM THE FOLLOWING PARAMETERS'
      write(6,*)'1. NUMBER OF HOURS OF RAIN'
      write(6,*)'2. ALL PRECIPITATION VALUES'
      write(6,*)'3. ONE PRECIPITATION VALUE'
      write(6,*)'4. EXIT TO MENU'
c
      k1=1
      read(5,*,iostat=k1,err=502)jr
      goto(285,286,505,435)jr
      goto 502
c
  505 write(6,*)'WHICH HOUR OF PRECIPITATION DO YOU WISH TO ALTER?'
      k1=1
      read(5,*,iostat=k1,err=505)i
      write(6,*)' INPUT PRECIPITATION (m) FOR HOUR ',i
      write(6,*)'PRESENT VALUE IS ',precip(i)
      k1=1
      read(5,*,iostat=k1,err=505)precip(i)
      goto 502
c
  503 write(6,*)'CHOOSE FROM THE FOLLOWING PARAMETERS'
      write(6,*)'1. INTERNAL ANGLES OF FRICTION, PHI AND PHIB'
      write(6,*)'2. SOIL COHESION'
      write(6,*)'3. SOIL SHEAR MODULUS'
      write(6,*)'4. DEPTH OF CONE TIP'
      write(6,*)'5. NUMBER OF SEGMENTS'
      write(6,*)'6. EXIT TO MENU'
c
      k1=1
      read(5,*,iostat=k1,err=503)jr
      goto(400,405,410,420,10,435)jr
      goto 503
c
 504  write(6,*)'CHOOSE FROM THE FOLLOWING PARAMETERS'
c
      write(6,*)'1. YI'
      write(6,*)'2. DEPMAX'
      write(6,*)'3. XI'
      write(6,*)'4. EXIT TO MENU'
c
c     write(6,*)' INPUT SEGMENT NO.'
c     read(5,*,iostat=k1,err=504)k
c     write(6,*)' INPUT N NO.'
c     read(5,*,iostat=k1,err=504)n
c
```

```
c
      k1=1
      read(5,*,iostat=k1,err=504)jr
      goto(601,602,603,435)jr
  601 write(6,*)' INPUT YI VALUE NO ',n,' SEGMENT ',k
      k1=1
      read(5,*,iostat=k1,err=504)yi(n,k)
      goto 504
  602 write(6,*)' INPUT DEPMAX VALUE NO. ',n,' SEGMENT NO. ',k
      write(6,*)'PRESENT VALUE IS ',depmax(n,k)
      k1=1
      read(5,*,iostat=k1,err=504)depmax(n,k)
      goto 504
  603 write(6,*)' INPUT XI VALUE NO. ',n, 'SEGMENT NO. ',k
      write(6,*)'PRESENT VALUE IS ',xi(n,k)
      k1=1
      read(5,*,iostat=k1,err=504)xi(n,k)
      goto 504
c
c
  100 format(1x,'THIS PART OF THE PACKAGE SETS UP AN INPUT FILE')
  110 FORMAT(/1X,'DO YOU WISH TO CONTINUE? y OR n')
c
c
c
 1010 nl=999
 1000 return
      end
```

```
c
c*****************************************************************
c
c SUBROUTINE    VOUTA
c
c*****************************************************************
c
      subroutine vouta
c
c
c This subroutine prints out the segment geometry,
c run information and hourly soil moisture information
c for the entire segment.
c Original by Bernier, 1982;Whitelaw, 1988.
c Adapted by Cochrane, 1988.
c
c
c 'vouta' is called by 'vcntl'
c
c 'vouta' uses functions 'vci'
c                         'vxmatrx'
c
c
      include 'spec.vgeom'
      include 'spec.vphys'
      include 'spec.vindex'
      include 'spec.vwater'
      include 'spec.vtime'
      include 'spec.vtitle'
      include 'spec.vkgr'
       include 'spec.vcom2'
c
      dimension pmc(5,16,5),satvol(5),sv75pc(5),totbl(5),
     2          totsm(5),total(5),totop(5),totbot(5),change(5),
     3          error(5),totsmp(5)
      dimension rci(3,16,5)
c
      entry vouta1
c
      WRITE (7, 100)
      WRITE (7, 110)
      WRITE (7, 120)NAME
      WRITE (7, 110)
      WRITE (7, 130)MNO
      WRITE (7, 110)
      WRITE (7, 140)IDAY,IMONTH,IYEAR
      WRITE (7, 110)
      WRITE (7, 150) NOTE
      WRITE (7, 110)
      WRITE (7, 100)
100   FORMAT(' ',100('*'))
110   FORMAT(' *',98(' '),'*')
120   FORMAT(' *',20(' '),15A4,18(' '),'*')
130   FORMAT(' *',20(' '),'TOTAL NUMBER OF SEGMENTS;',I4
     *      ,49(' '),'*')
140   FORMAT(' *',20(' '),'DATE OF EVENT (DAY/MONTH/YEAR);'
     *      ,I3,'/',I2,'/',I2,38(' '),'*')
150   FORMAT(' *',20(' '),18A4,6(' '),'*')
```

```
      RETURN
C
C
         entry vouta2
C
C
      do 112 k=1,kno
C
      WRITE (7, 200)ISGNO
      write (7, 205)k
      WRITE (7, 210)
      WRITE (7, 220)(N,N=1,16)
      WRITE (7, 230)(J,(avele(k,n,j),N=1,16),J=1,JNO)
      WRITE (7, 240)
      WRITE (7, 220)  (N,N=1,16)
      WRITE (7, 230)(J,(avslp(k,n,j),N=1,16),J=1,JNO)
      WRITE (7, 250)
      WRITE (7, 220)(N,N=1,16)
      WRITE (7, 230)(J,(avdep(k,n,j),N=1,16),J=1,JNO)
      WRITE (7, 260)
      WRITE (7, 220)(N,N=1,16)
      WRITE(7,230)(J,(avslln(k,n,j),N=1,16),J=1,JNO)
      WRITE(7,270)
      WRITE(7,220)(N,N=1,16)
      J=1
      WRITE (7, 230) J,(avwid(k,n),N=1,16)
      if(k.eq.kno) goto 112
      write(7,275)k,k+1
      write(7,220)(n,n=1,16)
      write(7,230)(j,(asslp(k,n,j),n=1,16),j=1,jno)
  112 CONTINUE
210   FORMAT(///' ELEVATION OF EACH SOIL ELEMENT (METRES)')
220   FORMAT('  J; N',I4,15I8)
230   FORMAT(' ',I2,1X,16F8.2)
240   FORMAT(/' SLOPE OF EACH SOIL ELEMENT ')
250   FORMAT(/' DEPTH OF EACH SOIL ELEMENT (METRES) ')
260   FORMAT(/' SLOPEWISE LENGTH OF EACH SOIL ELEMENT (METRES) ')
270   FORMAT(/' WIDTH OF EACH INCREMENT (METRES) ')
200   format(/1x,'HYDROGRAPH SIMULATION FOR SEGMENT No ',I3)
205   format(1x,'                        COMPONENT No ',i3)
  275 format(/1x,'SLOPES BETWEEN COMPONENT',i3,'  AND',i3)
      RETURN
C
C
C
      entry vouta3
C
C
      ix=ixm+1
C
      do 10 k=1,kno
            satvol(k)=0.
            sv75pc(k)=0.
      do 20 j=1,jno
             ni=nlimk(k)
      do 20 n=1,ni
            delpbv(k,n,j) = smcvol(k,n,j) / avvol(k,n,j)
            pmc(k,n,j)=delpbv(k,n,j)/poros(k,n,j)
            if(pmc(k,n,j).ge.0.750)sv75pc(k)=sv75pc(k)+avvol(k,n,j)
     2                                           *poros(k,n,j)
   20       if(pmc(k,n,j).ge.0.998)satvol(k)=satvol(k)+avvol(k,n,j)
     2                                           *poros(k,n,j)
c           poten(k,n,j)=vxmatrx(k,n,j,pmc(k,n,j))
   10 CONTINUE
c
c
      do 28 k=1,5
            totbl(k)=0.
            totsm(k)=0.
            total(k)=0.
            totop(k)=0.
            totbot(k)=0.
            error(k)=0.
            satvol(k)=0.
            sv75pc(k)=0.
   28 CONTINUE
      do 30 k=1,kno
            ni=nlimk(k)
      do 35 n=1,ni
            totop(k)=totop(k)+smcvol(k,n,1)
            totbot(k)=totbot(k)+smcvol(k,n,jno)
      do 35 j=1,jno
            totbl(k)=totbl(k)+avvol(k,n,j)
            totsm(k)=totsm(k)+smcvol(k,n,j)
   35 CONTINUE
      if(totbl(k).eq.0.)write(7,12345)k,n,j
12345 format('TOTBL = 0. k,n,j = ',3i3)
            total(k)=totsm(k)/totbl(k)
            change(k)=(totsm(k)-totsmp(k))/(-3.6)
            totsmp(k)=totsm(k)
            error(k)=eflow(k)-change(k)
   30 CONTINUE
c
c
c
      write(7,500)
      write(7,505)it,q
c      write(7,510)css,(flow(k,1),k=1,5),(totbl(k),k=1,5)
c      write(7,515)pnet,(flow(k,2),k=1,5),(totsm(k),k=1,5)
c      write(7,520)(flow(k,3),k=1,5),(totop(k),k=1,5)
c      write(7,525)(flow(k,4),k=1,5),(totbot(k),k=1,5)
c      write(7,530)(flow(k,5),k=1,5),(error(k),k=1,5)
c      write(7,540)(satvol(k),k=1,5)
c      write(7,545)(sv75pc(k),k=1,5)
c
      if(mod(it,npo).ne.0) goto 50
      write(7,550)(n,n=1,10)
      write(7,555)((j,(pmc(k,n,j),n=1,10),j=1,jno),k=1,kno)
      write(7,*)(((pmc(k,n,j),n=1,10),j=1,jno),k=1,kno)
      abai=0.0
      do 1000 n=1,10
      poten(1,n,1)=vxmatrx(1,n,1,pmc(1,n,1))
      write(7,*)poten(1,n,1)
      rci(1,n,1)=vci(poten(1,n,1))
      abai=abai+(rci(1,n,1)*avslln(1,n,1)*avwid(1,n))
 1000 CONTINUE
```

```
      abai=(abai/sgarea)
      write(12,*)abai
   50 CONTINUE
C
C
  500 format(/1x,130('-')/)
  505 format(1x,i4,' ##  ','Q ',f9.5,3x,'SS OUTFLOWS (l/s)',32x,
     &       'VOLUMES (m3)')
  510 format(1x,7x,'CSS ',f9.5,3x,'1',5f9.4,2x,5f8.2,1x,
     &       ' TOTAL VOL')
  515 format(1x,1x,8x,'P ',f6.2,6x,'2',5f9.4,2x,5f8.2,1x,
     &       ' WATER VOL')
  520 format(1x,23x,'3',5f9.4,2x,5(f7.1,1x),1x,' TOP VOL')
  525 format(1x,23x,'4',5f9.4,2x,5(f7.1,1x),1x,' BOT VOL')
  530 format(1x,23x,'5',5f9.4,2x,5(f7.1,1x),1x,' LOSS(l/s)')
  540 format(1x,71x,5(f7.3,1x),1x,' VOL SAT')
  545 format(1x,71x,5(f7.3,1x),1x,' VOL> 75%S')
  550 format(/3x,'  ',5x,'J ; N',10(i3,3x))
  555 format(3x,'##',4x,i2,4x,10f6.3)
C
C
      return
      end
```

```
c
c*****************************************************
c
      subroutine voutb
c
c
c This subroutine calculates the flow of water from the
c segment to the channel system and lags it accordingly.
c Original by Bernier, 1982;Whitelaw, 1988.
c
c
c 'voutb' is called by 'vcntl'
c
c
      INCLUDE 'spec.vgeom'
      INCLUDE 'spec.vtime'
      include 'spec.vwater'
c
      dimension tflow(500),rain(500),ecum(500,5)
c
      entry voutb1(isgno,it)
      do 10 k=1,5
   10 ecum(it,k) = eflow(k)

c
      RETURN
c
c
      entry voutb2(it,qi,prepi,prec)
      write(7,*)'******OUTB2********'
      INDEX=IT+(IROUT/60)
      IINDEX=INDEX+1
      LAG=MOD(IROUT,60)
      TFLOW(INDEX)=TFLOW(INDEX)+(1.-FLOAT(LAG)/60.)*QI
      TFLOW(IINDEX)=TFLOW(IINDEX)+(FLOAT(LAG)/60.)*QI
      RAIN(IT)=prec
C  ***Convert QI back to m3 before dividing by SQKM ***
      CMSK=(QI*3600.)/1000.
      IENTRY=2
      INDEX=IT
C
C
C
C
100             IF(INDEX.GT.1)GOTO 110
      RNFALL=0.
      ITIME=0.
      START=9999.
      STRFLO=0.
      STORM=0.
110   RNFALL=RNFALL+((RAIN(INDEX)*SQKM*1000000)*0.01)
      IF(RAIN(INDEX).EQ.0.)GOTO 120
      IF(START.NE.9999.)GOTO 120
      START=AMIN1(STRFLO,CMSK)
      IITIME=INDEX
120      STRFLO=CMSK
      IF(BASFLO.LE.CMSK)GOTO 130
      START=9999.
      ITIME=0.
      BASFLO=0.
      GOTO 140
C  ***Convert CMSK to mm before summing ***
130   STORM=STORM+(CMSK*1000)
140   IF(IENTRY.EQ.2)GOTO 400
      GOTO 300
C
C
      entry voutb3(it,mno)
      write(7,*)'++++++OUTB3++++++++'
      write(7,*)'^^^^^^OUTB3^^^^^^^^^'
C
      IIDAY=0.
      ITIME=0
      WRITE(7,601)
      WRITE(7,6015)
      DO 385 INDEX=1,IT
        IJ=IMIN+(INDEX*60)
        IMM=MOD(IJ,60)
        IHH=IJ/60
        ITIME=(IHOUR+IHH)*100+IMM
        ITIME=ITIME-(2400*IIDAY)
        IF(ITIME.GE.2400)IDAY=IDAY+1
        IF(ITIME.GE.2400)IIDAY=IIDAY+1
        IENTRY=3
        GOTO 100
300       CONTINUE
      write(7,602)index,rain(index),(ecum(index,k),k=1,5),tflow(index)
385   CONTINUE
400   CONTINUE
C
      RETURN
C
  601 format(//'   INDEX  RAIN',2x,
     & '-------- SUBSURFACE OUTFLOWS ---------  TOTAL')
 6015 format(10x,'(CM)',15x,'LITRES / SEC',14x,'(ROUTED)'/)
  602 format(3x,i3,2x,f6.2,6f8.4)
C605   FORMAT(/,'*THE STORMFLOW IN INCHES OF WATER IS',F15.4)
C606   FORMAT(/,'*THE RAINFALL IN INCHES OF WATER IS ',F15.4)
C607   FORMAT('1',' SEGMENT STORMFLOW(IN) REPONSE(%) ')
C608   FORMAT(I5,5X,F15.4,5X,F14.3)
C
      END
```

```
c
c*****************************************************
c
c SUBROUTINE     VREAD
c
c*****************************************************
c
      subroutine vread
c
c
c This subroutine reads the input data files for VSAS2.
c These are (1)segment, (2)storm, (3)initial moisture,
c and (4)soil strength characteristics.
c Original by Bernier, 1982;Whitelaw, 1988.
c Adapted by Cochrane, 1988.
c
c
c 'vread' is called by 'vcntl'
c                      'vmenu'
c
c
      include 'spec.vgeom'
      include 'spec.vphys'
      include 'spec.vindex'
      include 'spec.vwater'
      include 'spec.vtime'
      include 'spec.vdata'
      include 'spec.vtitle'
      include 'spec.vcom2'
      include 'spec.vkgr'
      include 'spec.vcone'
c
      entry vread1
c
      read(9,50)mno,name
50    FORMAT(i3,15a4)
      read(9,60)note
60    FORMAT(18a4)
      read(3,100)iday,imonth,iyear
100   FORMAT(3i2)
      read(9,91)krep,lrep,npo,nouta,ntmp
   91 FORMAT(5i3)
      i=0
10    i=i+1
      iii=i+7
      read(3,110)(precip(ii),ii=i,iii)
110   FORMAT(8f5.3)
      if(precip(i).ge.9.) return
c The 9.card marks the end of the precipitation deck
      i=iii
      goto 10
c
c
c
c
      entry vread2
c
      read(9,120)isgno,kno,jno,irout,sgarea
```

```
 120    FORMAT(3(i2,2X),I3,2X,F6.0)
        if(kno.ge.99)return
        read(4,90)(((pbv(k,n,j),n=1,10),j=1,jno),k=1,kno)
     90 FORMAT(6x,10f6.3)
        read(9,130)clevel,(stonec(j),j=1,jno)
 130    FORMAT(6f5.3)
        read(9,135)(nlimk(k),k=1,3)
 135    FORMAT(3i3)
        do 139 k=1,kno
         nn = nlimk(k)
         read(9,140)(yi(k,n),n=1,nn)
         read(9,140)(depmax(k,n),n=1,nn)
         read(9,140)(xi(k,n),n=1,nn)
  139    CONTINUE
 140    FORMAT(10f7.2)
        do 144 k=1,kno
         read(9,145)xwidth(k,1),xwidth(k,2),horiz(k,1),horiz(k,2)
  144    CONTINUE
 145    FORMAT(4f7.2)
        jj=jno-1
        read(9,150)adepth,(bdepth(j),j=1,jj),atest,cdepth
 150    FORMAT(7f6.3)
        read(9,170) a1,a2,(roffst(k),k=1,kno-1)
 170    FORMAT(f5.1,f10.1,4f7.2)
        read(9,180)(pos(k),k=1,3),(hlen(k),k=1,kno)
 180    FORMAT(3f4.1,5f7.2)
        do 300 k=1,kno
          read(9,190)(cmax1(k,j),poros1(k,j),ca1(k,j),cb1(k,j),j=1,jno)
          if(pos(1).ge.1.) goto 300
          read(9,190)(cmax2(k,j),poros2(k,j),ca2(k,j),cb2(k,j),j=1,jno)
          if(pos(1).ge.1.) goto 300
          read(9,190)(cmax3(k,j),poros3(k,j),ca3(k,j),cb3(k,j),j=1,jno)
    190 FORMAT(f12.6,f12.6,e12.3,f12.6)
    300 CONTINUE
c
        return
c
c
c
        entry vread3
        read(2,*)phi,c,phib,gg,zz
c
        aal=1.48
        d=0.799
        alpha=15.0
        gamma=0.067
        ua=-17.0
        return
        end
```

```
c
c****************************************************
c
c SUBROUTINE    VSURFLO
c
c****************************************************
c
      subroutine vsurflo(a1,a2,pnet,css)
C
c
c This subroutine calculates the surface runoff in VSaS2.
c Original by Bernier, 1982;Whitelaw, 1988.
c
c
c 'vsurflo' is called by 'vcntl'
c
c
C     a1=channel  a2=imp. surface area
C
      cI=a1*pnet/100.
      of=pnet*a2/100.
      css=ci+of
c
      return
      end
```

```
c
c****************************************************
c
c SUBROUTINE     VVIEW
c
c****************************************************
c
      subroutine vview
c
c
c This subroutine prints out the VSAS2 input files
c created or edited in vmenu.
c Created by J.E. Cochrane
c at the Geography Department, University of Bristol, England.
c December, 1987.
c
c
c 'vview' is called by 'vmenu'
c
c
      include 'spec.vgeom'
      include 'spec.vphys'
      include 'spec.vindex'
      include 'spec.vwater'
      include 'spec.vtime'
      include 'spec.vdata'
      include 'spec.vcom2'
      include 'spec.vkgr'
      include 'spec.vcone'
c
c
      write(9,50)mno,name
      write(9,60)note
      write(3,70)iday,imonth,iyear
      write(9,80)krep,lrep,npo,nouta,ntmp
      ival=(int((ir/8)+1))
      ival=(ival*8)+1
      precip(ival)=9.0
      ii=0
   10 ii=ii+1
      ir=ii+7
      write(3,90)(precip(i),i=ii,ir)
      if(precip(ii).lt.9.)goto 10
      write(9,110)isgno,kno,jno,irout,sgarea
      write(4,120)(((pbv(k,n,j),n=1,10),j=1,jno),k=1,kno)
c     write(9,130)clevel,(stonec(j),j=1,jno)
      write(9,130)clevel,(stonec(j),j=1,jno)
      write(9,140)(nlimk(k),k=1,3)
      do 142 k=1,kno
         nn=nlimk(k)
         write(9,150)(yi(k,n),n=1,nn)
         write(9,150)(depmax(k,n),n=1,nn)
         write(9,150)(xi(k,n),n=1,nn)
  142 CONTINUE
      do 155 k=1,kno
         write(9,160)xwidth(k,1),xwidth(k,2),horiz(k,1),horiz(k,2)
  155 CONTINUE
      jj=jno-1
      write(9,170)adepth,(bdepth(j),j=1,jj),atest,cdepth
      write(9,180)a1,a2,(roffst(k),k=1,kno-1)
      write(9,190)(pos(k),k=1,3),(hlen(k),k=1,kno)
      do 195 k=1,kno
         write(9,200)(cmax1(k,j),poros1(k,j),ca1(k,j),cb1(k,j),j=1,jno)
         if(pos(1).ge.1.)goto 195
         write(9,200)(cmax2(k,j),poros2(k,j),ca2(k,j),cb2(k,j),j=1,jno)
         if(pos(1).ge.1.)goto 195
         write(9,200)(cmax3(k,j),poros3(k,j),ca3(k,j),cb3(k,j),j=1,jno)
      write(9,*)'01999999999999999999999999999999999999'
  195 CONTINUE
         write(2,*)phi,c,phib,gg,zz
c
c
   50 format(i3,15a4)
   60 format(18A4)
   70 format(3I2)
   80 format(5i3)
   90 format(8f5.3)
  110 format(3(i2,2x),i3,2x,f6.0)
  120 format(6x,10f6.3)
  130 format(6f5.3)
  140 format(3i3)
  150 format(10f7.2)
  160 format(4f7.2)
  170 format(7f6.3)
  180 format(f5.1,f10.1,5f7.2)
  190 format(3f4.1,5f7.2)
  200 format(f12.6,f12.6,f12.3,f12.6)
c
c
      return
      end
```

```
c
c*****************************************************
c
c FUNCTION     VXMATRX
c
c*****************************************************
c
      function vxmatrx(k,n,j,pb)
c
c
c This function calculates the matric potential
c according to Cambell's method.
c Original by Bernier, 1982; Whitelaw, 1988.
c
c
c 'vxmatrx' is called by 'vdrain'
c                        'vouta'
c
c
      include 'spec.vphys'
      include 'spec.vindex'
c
c
      cmult    = ((pb/poros(k,n,j))**cb(k,n,j))
      xmat = (-1.) * ca(k,n,j) * ( 1. / cmult )
      vxmatrx=amin1(0.,xmat)
c
      return
      end
```

APPENDIX C : Compilation Instructions for the BORTS

1. Compile with FORTRAN 77 compiler according to.-

```
f77 -c *.f
f77 -o master *.o -lnag
```

Notes: 'lnag' is the NAG routine option

'master' will be the file to be run in section 10